美国数学会经典影印系列




# Algebraic Curves and Cryptography

## 代数曲线与密码学

Editor: V. Kumar Murty

高等教育出版社

美国数学会经典影印系列




# Algebraic Curves and Cryptography

## 代数曲线与密码学

Editor: V. Kumar Murty

高等教育出版社 · 北京

图字：01-2018-2922号



**代数曲线与密码学**

Daishu Quxian yu Mimaxue

**图书在版编目 (CIP) 数据**

代数曲线与密码学 = Algebraic Curves and Cryptography : 英文 / ( 加 ) 库马尔 · 默蒂 (V. Kumar Murty) 主编 . -- 影印本 . -- 北京 : 高等教育出版社 , 2019.1
ISBN 978-7-04-051038-6
Ⅰ. ①代… Ⅱ. ①库… Ⅲ. ①代数曲线－英文 ②密码学－英文 Ⅳ. ①O187.1 ②TN918.1
中国版本图书馆 CIP 数据核字 (2018) 第 274989 号

策划编辑　李华英　　责任编辑　李华英
封面设计　张申申　　责任印制　陈伟光

出版发行 高等教育出版社
社址 北京市西城区德外大街4号
邮政编码 100120
购书热线 010-58581118
咨询电话 400-810-0598
网址 http://www.hep.edu.cn
http://www.hep.com.cn
网上订购 http://www.hepmall.com.cn
http://www.hepmall.com
http://www.hepmall.cn
印刷 北京新华印刷有限公司

开本 787mm×1092mm 1/16
印张 9.25
字数 230千字
版次 2019年1月第1版
印次 2019年1月第1次印刷
定价 67.00元


[物 料 号 51038-00]



美国数学会经典影印系列

# 出版者的话

近年来，我国的科学技术取得了长足进步，特别是在数学等自然科学基础领域不断涌现出一流的研究成果。与此同时，国内的科研队伍与国外的交流合作也越来越密切，越来越多的科研工作者可以熟练地阅读英文文献，并在国际顶级期刊发表英文学术文章，在国外出版社出版英文学术著作。

然而，在国内阅读海外原版英文图书仍不是非常便捷。一方面，这些原版图书主要集中在科技、教育比较发达的大中城市的大型综合图书馆以及科研院所的资料室中，普通读者借阅不甚容易；另一方面，原版书价格昂贵，动辄上百美元，购买也很不方便。这极大地限制了科技工作者对于国外先进科学技术知识的获取，间接阻碍了我国科技的发展。

高等教育出版社本着植根教育、弘扬学术的宗旨服务我国广大科技和教育工作者，同美国数学会（American Mathematical Society）合作，在征求海内外众多专家学者意见的基础上，精选该学会近年出版的数十种专业著作，组织出版了“美国数学会经典影印系列”丛书。美国数学会创建于1888年，是国际上极具影响力的专业学术组织，目前拥有近30000会员和580余个机构成员，出版图书3500多种，冯·诺依曼、莱夫谢茨、陶哲轩等世界级数学大家都是其作者。本影印系列涵盖了代数、几何、分析、方程、拓扑、概率、动力系统等所有主要数学分支以及新近发展的数学主题。

我们希望这套书的出版，能够对国内的科研工作者、教育工作者以及青年学生起到重要的学术引领作用，也希望今后能有更多的海外优秀英文著作被介绍到中国。

高等教育出版社

2016年12月

# Contents

Fields Institute Communications
Volume **58**, 2010

CHAPTER 1

# An Overview of Algebraic Curves and Cryptography

V. Kumar Murty

**Abstract.** This volume is based on seminars held at the GANITA Lab during 2001-2008. A few of the talks have been selected to present the themes of point counting and arithmetic on algebraic curves and Abelian varieties over finite fields. The presentations are mostly suitable for independent study by graduate students who wish to enter the field, both in terms of introducing basic material as well as guiding them in the literature.

## 1.1 Introduction

In 2001, the GANITA Lab was founded to study the applications of mathematics to problems in Information Technology. The Samskrit word *ganita* means computation or calculation and this is certainly one of the themes of the research activities in the lab. The name GANITA is also an acronym standing for Geometry, Algebra, Number Theory and their Information Technology Applications.

Over the past 8 years, the lab has mostly concentrated on applications related to information security. Part of the mandate of the lab was to contribute to the training of students and postdoctoral fellows who wanted to enter the area. For this purpose, a weekly seminar was held to discuss background material as well as to learn about recent research. This volume is a small selection of some of those seminar talks. They are arranged around the theme of point counting on various classes of Abelian varieties over finite fields.

In this introductory article, we will describe the setting, introduce some of the work that has been done in these areas, and outline the contents of the remaining articles in this volume.

## 1.2 The basic paradigm

Let us begin by recalling the basic paradigm of group based cryptography. We have a finite group $G$ and a distinguished element $g \in G$. We wish to use this information to establish a shared secret between two parties communicating over an unsecure channel. Each of the parties chooses an integer, say $x$ and $y$ and transmits $g^x$ (resp. $g^y$) to the other party. From this information, both parties can compute $g^{xy}$ and this is the shared secret. This is mathematically secure provided $g^{xy}$ cannot be deduced from $g^x$ and $g^y$. When we say that it "cannot" be deduced, we mean that it cannot be done in a computationally feasible manner.

Before proceeding, it may be worthwhile to make a few remarks about the meaning of security. As is well-known, the famous and oft-quoted protocol of the last paragraph is not used in practice because it is in fact not secure. It is susceptible to an attack by an adversary who stands in the line of communication and pretends to each of the legitimate parties that it is the other legitimate party. This attack, however, has nothing to do with a weakness in the underlying mathematical algorithm or the assertion about computational infeasibility. Rather, it has to do with the protocol.

Thus, the assertion that something is secure or is not secure without further qualification is likely to be a meaningless statement. Information security has many components (such as physical security, etc.) and many layers (such as mathematical, algorithmic, protocol layers, etc.) This volume deals exclusively with some aspects of mathematical and algorithmic security.

## 1.3 The Diffie-Hellman decision problem

Given a finite cyclic group $G$ with generator $g$, this problem asks whether given $g^x$ and $g^y$, we can deduce the value of $g^{xy}$. There are computational and mathematical reasons for expecting that if the group $G$ is chosen properly, the three quantities tend to behave independently. Indeed, one way of measuring independence is through the estimation of exponential sums. For $G = \mathbb{F}_p^\times$, there are results of Canetti et al. [16] about the size of

$$\sum_{x,y\in\mathbb{F}_p^\times} \exp\{ag^x + bg^y + cg^{xy}\}$$

for general $a, b, c$. The trivial bound is $p^2$ and any improvement of this is an indication that there is a degree of independence of distribution. The above paper proves that the sum is $\mathbf{O}(p^{23/12})$. The work of Canetti et al. builds on earlier work of Bourgain [13]. There are also elliptic analogues of this by Lange and Shparlinski [56]. These results give some grounds for believing that triples $\{g^x, g^y, g^{xy}\}$ cannot be deduced from random triples by their statistical properties. It is still worthwhile to ask if any partial information about $g^{xy}$ can be obtained. For example, a result of Boneh and Venkatesan [12] states that if the $\mathbf{O}(\sqrt{\log p})$ most significant bits of $g^{xy}$ can be computed from $g, g^x$ and $g^y$, then this is enough to deduce the entire value of $g^{xy}$.

## 1.4 Constraints on the group

Let $N$ denote the order of $G$. Trial and error would require about $\mathbf{O}(N)$ group calculations to determine $x$ and $y$ given $g^x$ and $g^y$. A generic algorithm called the Baby Step-Giant Step method could reduce this to $\mathbf{O}(\sqrt{N})$ group calculations. If we can choose the group $G$ so that there is no known algorithm that can solve the problem in fewer group calculations, the system will be considered to be computationally secure.

Firstly, since all calculations are taking place in the subgroup generated by $g$, we may as well assume that $G$ is cyclic and generated by $g$. Moreover, if $N$ is not prime or nearly prime, the Chinese Remainder Theorem can be used to reduce the number of calculations below the square root bound. The reader is referred to [11] for an expanded discussion of these considerations.

In order to verify that the selected group $G$ satisfies these considerations, it seems necessary to compute the order $N$. The point counting problem is exactly this.

This raises several interesting computational problems in finite groups. Given a finite Abelian group $G$ and an element $g \in G$, how can one compute the order of $g$? How can one compute the structure of $G$ (in other words, its decomposition as a product of cyclic groups)? What can one say if the group is not Abelian? See [77] for some work on solvable permutation groups. Also, can the question of whether the order of a group is prime or not be answered without actually computing the order?

## 1.5 Abelian varieties over finite fields

Abelian varieties over finite fields are a source of finite groups that might be used for discrete log based cryptographic systems. Let $A$ be an Abelian variety over a finite field $\mathbb{F}$. The possibility of using the group $A(\mathbb{F})$ of points on $A$ in $\mathbb{F}$ as the basis of cryptography is still at an early stage of exploration, though special families of Abelian varieties have been extensively studied. The most attention has been given to elliptic curves and to the Jacobians of hyperelliptic curves. There are now many reference works on these two families of curves [61], [19].

Denote by $\overline{\mathbb{F}}$ an algebraic closure of $\mathbb{F}$ and let

$$G = \mathrm{Gal}(\overline{\mathbb{F}}/\mathbb{F})$$

be the Galois group. Since $\mathbb{F}$ has a unique extension of degree $n$ for each $n$, $G$ is a procyclic group:

$$G \simeq \hat{\mathbb{Z}} \simeq \mathrm{proj}\lim \mathbb{Z}/n\mathbb{Z}.$$

Let $\mathrm{Frob} = \mathrm{Frob}_{\mathbb{F}}$ be the map

$$x \mapsto x^q$$

where $q$ is the number of elements in $\mathbb{F}$. Sometimes, we may also write $\mathrm{Frob}_q$. It is a topological generator of $G$.

There is an action of $G$ on $A(\overline{\mathbb{F}})$. In particular, the function

$$n \mapsto \deg(\mathrm{Frob} - n)$$

is well defined. There is a polynomial $P_A(T)$ with the property that for every $n$ (sufficiently large),

$$P_A(n) = \deg(\mathrm{Frob} - n).$$

This is called the characteristic polynomial of the Frobenius automorphism. It has the property that

$$|A(\mathbb{F})| = P_A(1).$$

Moreover, if $d$ is the dimension of $A$,

$$P_A(T) = \prod_{i=1}^{2d} (1 - \omega_i T)$$

where

$$|\omega_i| = q^{\frac{1}{2}}$$

and for $1 \leq i \leq d$,

$$\omega_i \omega_{d+i} = q.$$

We see from this that

$$|A(\mathbb{F})| = q^d + \mathbf{O}(q^{d-\frac{1}{2}}). \tag{1.1}$$

## 1.6 Elliptic curves

A one-dimensional Abelian variety is an elliptic curve. It is a curve of genus one on which an algebraic group law can be defined. Amongst all curves, it is only those of genus one for which such a group law exists. The advantage of elliptic curves is that the equation that defines such a curve and the equations that define the group law can be given in explicit and elementary terms.

If $E$ is an elliptic curve defined over the finite field $\mathbb{F}$ of $q$ elements, then

$$\#E(\mathbb{F}_q) = q + 1 - a_q$$

for some integer $a_q$ that satisfies $|a_q| \leq 2\sqrt{q}$. This one integer in fact determines the number of points on $E$ over any extension field of $\mathbb{F}_q$. Indeed, if we denote by $\alpha_q$ and $\beta_q$ the roots of the quadratic equation

$$T^2 - a_q T + q = 0$$

Then,

$$\#E(\mathbb{F}_{q^n}) = q^n + 1 - (\alpha_q^n + \beta_q^n).$$

The problem of computing the group order $\#E(\mathbb{F}_q)$ is therefore equivalent to computing the function

$$Z(E,T) = \exp\{\sum_{n=1}^{\infty} \#E(\mathbb{F}_{q^n})T^n/n\}.$$

This function, called the zeta function of $E$, is known to be a rational function of the form

$$\frac{P_E(T)}{(1-T)(1-qT)}$$

where $P_E(T)$ is a polynomial with integer coefficients. In fact,

$$P_E(T) = T^2 - a_q T + q.$$

This is the polynomial referred to in the previous section and in particular, $P_E(1) = \#E(\mathbb{F}_q)$.

Techniques for computing zeta functions for more general curves can be found in the papers of Lauder and Wan [57], [58], [87]. We note that the techniques of these papers are only effective for small characteristic. For large characteristic, the problem is still unsolved.

## 1.7 Statistical results

Once we have solved the computational problem of determining the number of points, there is the question of whether this number is nearly prime. From one point of view, we may regard this as a primality testing problem, and apply known algorithms.

It is also an interesting question to ask for the likelihood that the number of points is prime or nearly prime. The question can be made precise in several ways. Firstly, we may consider the collection of elliptic curves over $\mathbb{F}_p$ or $\mathbb{F}_q$ and ask what proportion of them have a prime number of points. This question in turn requires us to examine the possible group orders that can occur.

For an elliptic curve $E$ over $\mathbb{F}_p$, the group order satisfies

$$p + 1 - 2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}.$$

For $p \neq 2$, every integer in this interval does actually occur as the group order for some elliptic curve over $\mathbb{F}_p$. Denote by $P$ the probability that a number in this interval is prime. If primes are equidistributed in this interval, we might expect that it contains asymptotically $4\sqrt{p}/\log p$ primes and so $P = 1/\log p$. Galbraith and McKee [30] have proposed some heuristics that suggest that the probability that a randomly chosen elliptic curve over $\mathbb{F}_p$ has a prime group order is $c_p P$ where $c_p$ is given by the infinite product

$$c_p = \frac{2}{3}\prod_{\ell>2}\left(1-\frac{1}{(\ell-1)^2}\right)\prod_{2<\ell|(p-1)}\left(1-\frac{1}{(\ell+1)(\ell-2)}\right).$$

Numerically, $c_p$ lies between 0.44 and 0.62. Thus, if one were looking for an elliptic curve over a field of size $p \sim 2^{256}$ with prime group order, then according to this conjecture, one would have to test approximately 358 random curves.

This conjecture is still open. Both the conjecture and its variants for hyperelliptic curves are probably a fruitful area of study.

Another way to approach the question statistically is to begin with an elliptic curve $E$ over the rational numbers $\mathbb{Q}$ and to look at reductions modulo various primes $p$. Here, analytic and sieve methods are able to shed some light on what happens as $p$ varies.

There is the following result of Miri and Murty [63]. Let $E$ be an elliptic curve over $\mathbb{Q}$. Assuming the Riemann Hypothesis (for all Dedekind zeta functions), we have that

$$|E(\mathbb{F}_p)|$$

has $\log\log p$ prime divisors for a set of primes of density 1. In the case that $E$ has complex multiplication, this has been proved unconditionally and independently by Cojocaru [20] and by Liu[59]. Since $\log\log p$ grows *very* slowly with $p$, these results say that the order $|E(\mathbb{F}_p)|$ has essentially a bounded number of prime factors when $p$ is in cryptographic ranges.

Note that there is a conjecture of Koblitz [50] that asserts that $|E(\mathbb{F}_p)|$ should be prime for

$$\sim c_E \frac{x}{(\log x)^2}$$

of the primes $p \leq x$. Here $c_E$ is a constant that is described as an infinite product over all primes, with each factor contributing a "local density". He made this conjecture in analogy with the conjectures of Hardy and Littlewood about primes of the form $2p+1$. Koblitz's conjecture is still open. The first results on this question were in the work of Miri and Murty [63]. There it was shown that assuming the GRH, there are

$$\gg \frac{x}{(\log x)^2}$$

primes $p \leq x$ such that $|E(\mathbb{F}_p)|$ has at most 13 prime divisors.

This used the lower bound Selberg sieve method. In the case that $E$ has complex multiplication, Cojocaru [20] has improved the bound from 13 to 5 and has removed the assumption of the GRH by invoking an analogue of the Bombieri-Vinogradov theorem.

We can also ask about large prime divisors of $|E(\mathbb{F}_p)|$. Recent work of Akbary and Murty [1] gives information on this problem. Let $\Gamma$ be a free subgroup of rank $r$ (say) in the group of rational points $E(\mathbb{Q})$ of $E$. Reducing this subgroup modulo $p$ gives us a subgroup $\Gamma_p$ (say) of $E(\mathbb{F}_p)$. From the ground breaking work of Gupta

and R. Murty [39] it is known that for any $\epsilon > 0$, there is a set of primes $p$ of density 1 for which we have

$$|\Gamma_p| \geq p^{1-\frac{2}{r+2}-\epsilon}.$$

If $E$ is a prime with complex multiplication, we prove in [1] that this can be slightly improved to

$$|\Gamma_p| \geq p^{1-\frac{2}{r+2}+\epsilon(p)}$$

where $\epsilon(p)$ is any function such that $\epsilon(p) \longrightarrow 0$ as $p \longrightarrow \infty$. This is the elliptic analogue of results of Erdös and R. Murty [26] for the group $\mathbb{G}_m$.

Moreover, in Akbary, Ghioca and Murty [2], under different hypotheses, stronger lower bounds are given for the $|\Gamma_p|$. More precisely, let us suppose that the rank $r$ of $\Gamma$ is atleast 11. Also, assume that Artin $L$-functions are analytic (except possibly at $s = 1$), and that the Riemann Hypothesis holds for Dedekind zeta functions. Then for a set of primes $p$ of density 1, we have

$$|\Gamma_p| \geq p/f(p)$$

where $f$ is any function such that $f(p) \longrightarrow \infty$ as $p \longrightarrow \infty$.

Since $\Gamma_p$ is a subgroup of $E(\mathbb{F}_p)$ and this latter group has order $p + \mathbf{O}(\sqrt{p})$, the above result is saying that almost all of $E(\mathbb{F}_p)$ comes from reduction modulo $p$ of global points. Whether this condition has cryptographic implications is not clear and is worthy of further study.

## 1.8 Abelian varieties of higher dimension

Since Abelian varieties of higher dimension have more points (roughly $q^d$ where $d$ is the dimension), a generic attack on the Discrete Logarithm Problem should take about

$$q^{d/2}$$

steps. This means that it may be possible to use them as the basis of a secure cryptographic scheme with a smaller value of $q$. In other words, the mathematical security provided by an Abelian surface ($d = 2$) over a field of $q = 2^{83}$ elements should be comparable to the level of mathematical security provided by an elliptic curve ($d = 1$) over a field of $q = 2^{163}$. The use of a smaller field means that field arithmetic should be simpler and faster. As field arithmetic underlies arithmetic on the Abelian variety itself, one might hope that calculations can be done in less time. The problem, of course, is that the arithmetic on higher dimensional Abelian varieties, to the extent that this can be made explicit, tends to be more complicated. Thus, there is a tradeoff between the speedup of field arithmetic and the cost of more complicated Abelian variety arithmetic.

For explicit and efficient arithmetic, most effort has been directed at elliptic curves. Some of this is, in fact, about improving the efficiency of arithmetic in finite fields. These results can be applied directly in the higher dimensional case as well.

In the case of Abelian varieties that are Jacobians of hyperelliptic curves, there has been significant progress in developing efficient arithmetic. There has also been progress on the arithmetic of Abelian varieties that arise as the Jacobian of more general curves. The class of $C_{ab}$ curves was considered by Arita [5] and developed in work of Arita, Miura and Sekiguchi [6]. The $C_{ab}$ curves are described in detail in this volume.

For Abelian varieties that are Jacobians of hyperelliptic curves of genus 3, the work of Guyot, Kaveh and Patankar [40] shows that in some cases, the arithmetic is faster than comparable elliptic curve arithmetic. Their work builds on the explicit formula method of Tanja Lange and others. Alas, there is also a new attack on these same Abelian varieties, due to Thériault [82] and refined by Gaudry et al. [36]. See also the attack of Diem and Thomé [24] which under a heuristic assumption casts doubt on the usefulness of any non-hyperelliptic curve of genus 3.

For the problem of point counting, there are fast algorithms in the case of hyperelliptic Jacobians over fields of small characteristic (work of Satoh [72], Fouquet-Gaudry-Harley [27], Kedlaya [48], Denef-Vercauteren [23], and others). For the case of a general Abelian variety, there is only a baby step-giant step approach to point counting. (See Gaudry [35], Chao, Matsui and Tsujii [18] and Izadi and Murty [45]).

One has also to consider whether there are new attacks possible in the higher dimensional case that somehow were not visible in the one dimensional case. We mention two examples of this.

Gaudry [33] showed that for Jacobians of hyperelliptic curves of genus $\geq 4$ over a field of $q$ elements, the discrete logarithm problem can be solved in time

$$\mathbf{O}_g(q^{2+\epsilon}).$$

Here, the implied constant depends on the genus $g$ of the curve. In fact, Gaudry's argument shows that the dependence is proportional to $g!$. Moreover, the $\epsilon$ in the exponent can be replaced by a small power of $\log q$. Note that the generic attack would require $g^{g/2}$ steps and so this is an improvement for $g \geq 5$. Thériault has shown how to extend this attack to $g \geq 3$. The work of Smith [78, 79] shows how to use explicit isogenies to transfer the discrete logarithm problem from certain genus 3 hyperelliptic curves to certain genus 3 non-hyperelliptic curves, where they are vulnerable to index calculus attacks.

This suggests that Jacobians of hyperelliptic curves of genus $g \geq 5$ may not be optimal for cryptography. However, as shown by Frey (see [19]), under some circumstances, this attack also affects curves of lower genus. In Frey's version, one begins with an Abelian variety $A$ of dimension $d$ (say) over a field $\mathbb{F}_{q^{mn}}$. One then considers the restriction of scalars $R_{\mathbb{F}_{q^{mn}}/\mathbb{F}_{q^m}} A$. This is an Abelian variety of dimension $dn$ defined over $\mathbb{F}_{q^m}$. Under some circumstances, this Abelian variety can be shown to be the Jacobian of a hyperelliptic curve of genus $dn$. As long as $dn \geq 3$, the index calculus attacks of Gaudry [33] and Thériault can be applied to this Jacobian and the discrete logarithm problem can be solved in time significantly faster than that of the generic method. An explicit example of this attack is given in the paper of Jacobson, Menezes and Stein [46].

Another instance of new problems that emerge in the higher dimensional case is the recent work of Murty and Patankar [69].

## 1.9 Outline of contents

This volume contains five articles. The first, by N. Thériault, gives an exposition of Schoof's method of counting the number of points on an elliptic curve over a finite field. It is an $\ell$-adic method, in the sense that for many primes $\ell \neq p$, where $p$ is the characteristic of the finite field, the number of points is computed modulo $\ell$. The Chinese remainder theorem is then used to deduce the actual number of points. This article gives detailed background on elliptic curves and on the basic

Schoof method. The Atkin-Elkies refinements are mentioned but not derived in detail. Issues related to implementation are also discussed. In particular, pseudo-code is provided. In the final section of the article, the possibility of performing the computations in parallel, rather than serially, is discussed.

Recent work of F. Morain takes the study of Schoof's algorithm further by considering computations modulo powers of primes and also by exploiting graphs of isogenies between elliptic curves. This work is not discussed in the present volume and the interested reader is referred to [68, 28], and related papers for the details.

In contrast to Schoof's $\ell$-adic method, Satoh [72] proposed a $p$-adic method to count the number of points on an elliptic curve over a finite field. This involved lifting the elliptic curve, together with the Frobenius endomorphism, to characteristic zero. The method works well for finite fields in which the number of elements is large compared to the characteristic.

Soon after the publication of Satoh's method, a variant suitable for Jacobians of hyperelliptic curves was proposed by Kedlaya [48]. This method used $p$-adic cohomology of Monsky and Washnitzer. This cohomology theory was a precursor to Grothendieck's crystalline cohomology and Berthelot's rigid cohomology. Kedlaya's work was for odd characteristic and it was extended to even characteristic by Denef and Vercauteren [23]. The second article is by Z. Ashraf, A. Juma and P. Sastry and gives a brief account of the method of Denef and Vercauteren. Again, implementation issues are discussed and pseudo-code provided for all the routines.

In the remaining three articles, we discuss $C_{ab}$ curves. This family of curves includes elliptic, and some hyperelliptic and superelliptic curves. In particular, they include the so-called "imaginary" hyperelliptic curves. These curves are characterized by the property that they have only one point at infinity.

The fourth article, by M. Radi-Benjelloun, gives an exposition of the theory of Gröbner bases, a tool from commutative algebra that is very useful in working with $C_{ab}$ curves. The idea of Gröbner bases is to generalize to multivariate polynomials the following well-known and elementary fact for polynomials of one variable. Given an ideal $I$ in $\mathbb{F}_q[x]$, there is a polynomial $f$ so that any other polynomial $g$ is in $I$ if and only if it is divisble by $f$. Given an ideal $I$ in $\mathbb{F}_q[x_1, \cdots, x_n]$, a Gröbner basis is a finite set of polynomials $S$ that generates $I$ and so that a general polynomial $g$ is in $I$ if and only if the remainder of division of $g$ by each element of $S$ is zero. Such a basis exists and by Buchberger's algorithm, can be efficiently computed. Though an ideal may have many Gröbner bases, there is a concept of a "reduced" Gröbner basis that is essentially uniquely determined by the ideal.

The next article, by F. Izadi, applies the theory of Gröbner bases to study arithmetic on the Jacobian of $C_{ab}$ curves. The elements of the Jacobian may be represented as ideal classes in the function field of the curve and reduced Gröbner bases are used to define the arithmetic.

The final article, by K. Shum, describes two towers of curves and the computation of their zeta functions. In particular, the number of points on these curves is computed. Both families are examples of $C_{ab}$ curves. The arithmetic on the Jacobian of general curves has been studied by several authors including Hess [43], Khuri-Makdisi [53], Arita, Miura and Sekiguchi [6], and Volchek [85].

The literature in cryptography seems to be growing at an exponential (perhaps doubly exponential!) rate. For a new entrant into the subject, navigating through this ocean can seem quite daunting. In this volume, the reader is steered toward a discussion of a few key ideas of the subject, together with some brief guidance

for further reading. It is hoped that this approach may render the subject more approachable.

**V. Kumar Murty**
Department of Mathematics
University of Toronto
Toronto, ON, Canada, M5S 2E4
murty@math.toronto.edu



Fields Institute Communications
Volume **58**, 2010

CHAPTER 2

# Schoof's Point Counting Algorithm

NICOLAS THÉRIAULT

**Abstract.** In this chapter, we will see how the algorithm of Schoof allows us to count the number of points of an elliptic curve defined over a finite field of large characteristic in polynomial time. After some essential background on elliptic curves (Sections 2.1 and 2.2), we will present the algorithm itself in Section 2.3. This will be followed in Section 2.4 by a detailed discussion on the implementation aspects necessary to make the algorithm more efficient in practice.

In Section 2.5 we present the improvements brought in by Atkin and Elkies to reduce the running time of Schoof's algorithm (giving the complete SEA algorithm). In the following sections (2.6 to 2.8), we describe the principal implementation aspects that are required to use these improvements.

## 2.1 Preliminaries

Before we see how to compute the group order of elliptic curves over prime fields, we need to establish a few results that will be used throughout the text.

**2.1.1 Elliptic curves over $\mathbb{C}$.** Over the complex numbers, an elliptic curve $E$ is isomorphic to a torus $\mathbb{C}/L$ where $L$ is the lattice $\omega_1\mathbb{Z}+\omega_2\mathbb{Z} \subset \mathbb{C}$ with $\omega_1/\omega_2 \notin \mathbb{R}$ (i.e. $\omega_1$ and $\omega_2$ must be linearly independent over $\mathbb{R}$).

For simplicity, we will assume that there is no lattice point in the parallelogram defined by $\{0, \omega_1, \omega_2, \omega_1 + \omega_2\}$. Every element of $\mathbb{C}/L$ can then be represented uniquely in the form $z \equiv r\omega_1 + s\omega_2$ with $r, s \in \mathbb{R}$, $0 \leq r, s < 1$.

To obtain the equation of the curve, we have to look at the Weierstraß function

$$\mathcal{P}(z) = \frac{1}{z^2} + \sum_{\omega \in L\backslash 0} \left( \frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right) ,$$

which is an elliptic function, i.e. $\mathcal{P}(z)$ is meromorphic and $L$-periodic. The Laurent series of $\mathcal{P}(z)$ around 0 is

$$\mathcal{P}(z) = \frac{1}{z^2} + \sum_{n=1}^{\infty} (2n+1) s_{2n+2} z^{2n} ,$$

where the $s_m$ are defined as $s_m = \sum_{\omega \in L\backslash 0} \frac{1}{\omega^m}$. The first and second derivatives of $\mathcal{P}(z)$ (which must also be elliptic functions) are

$$\mathcal{P}'(z) = \frac{-2}{z^3} + \sum_{n=1}^{\infty} (4n^2+2n) s_{2n+2} z^{2n-1}$$

and

$$\mathcal{P}''(z) = \frac{6}{z^4} + \sum_{n=1}^{\infty} (8n^3 - 2n)s_{2n+2}z^{2n-2} \ .$$

**Proposition 2.1** *If we set $a = -15s_4$ and $b = -35s_6$, then we have*

$$\left(\frac{1}{2}\mathcal{P}'(z)\right)^2 = (\mathcal{P}(z))^3 + a\mathcal{P}(z) + b$$

*and*

$$\mathcal{P}''(z) = 6(\mathcal{P}(z))^2 + 2a$$

**Proof** Since $\mathcal{P}$, $\mathcal{P}'$, and $\mathcal{P}''$ are meromorphic and $L$-periodic, the identities must also be meromorphic and $L$-periodic functions of $z$. To show the identities (written so the right-hand term is 0), it is therefore sufficient to show that they have no poles (so they must be constant, by Liouville's Theorem) and their constant term is zero. If we consider the $z$-expansion of the terms in the first equation, we have

$$\begin{aligned}
\left(\frac{1}{2}\mathcal{P}'(z)\right)^2 &= \frac{1}{z^6} - \frac{6s_4}{z^2} - 20s_6 + O(z^2) \\
-(\mathcal{P}(z))^3 &= -\frac{1}{z^6} - \frac{9s_4}{z^2} - 15s_6 + O(z^2) \\
-a\mathcal{P}(z) &= \frac{15s_4}{z^2} + O(z^2) \\
-b &= 35s_6 \ .
\end{aligned}$$

Since the terms on the right hand side add to $O(z^2)$, the sum must be zero, giving us the desired identity. Similarly, for the second identity we find

$$\begin{aligned}
\mathcal{P}''(z) &= \frac{6}{z^4} + 6s_4 + O(z^2) \\
-6(\mathcal{P}(z))^2 &= -\frac{6}{z^4} - 36s_4 + O(z^2) \\
-2a &= 30s_4 \ ,
\end{aligned}$$

and once again the right hand side must add up to zero. □

From now on, we let $X = \mathcal{X}(z) = \mathcal{P}(z)$ and $Y = \mathcal{Y}(z) = \frac{1}{2}\mathcal{P}'(z)$. With this notation, the first relation of Proposition 2.1 becomes

$$Y^2 = X^3 + aX + b \ , \tag{2.1}$$

and the elliptic curve $E$ is the set of points $(X, Y)$ satisfying Equation 2.1, with an extra point $P_\infty$, the *point at infinity.* The map $z \mapsto (\mathcal{X}(z), \mathcal{Y}(z))$ gives us a bijection between the points $z \in \mathbb{C}/L$ and the points of $E$ (in this map, $P_\infty$ is the image of the points of the lattice $L$). For any point $P$ on $E$, we have the two functions $\mathcal{X}(P)$ and $\mathcal{Y}(P)$ which return the $x$ and $y$ coordinates of $P$.

We also introduce two concepts that will be useful in Section 2.5 onward:

**Definition 2.2** An elliptic curve $E$ given by the equation $Y^2 = X^3 + aX + b$ has discriminant $\Delta = 27b^2 + 4a^3$ and $j$-invariant $j = 1728\frac{4a^3}{\Delta}$.

We will not use the $j$-invariant until Section 2.5, but the discriminant will play a small background role due to the following proposition:

**Lemma 2.3** *The equation $Y^2 = X^3 + aX + b$ defines an elliptic curve $E$ if and only if $\Delta = 27b^2 + 4a^3 \neq 0$ (if and only if $X^3 + aX + b = 0$ has no double root; if and only if $\gcd(X^3 + aX + b, 3X^2 + a) = 1$).*

In general, we will use the equation $Y^2 = X^3 + aX + b$ to define the elliptic curve $E$ and we will assume the properties of the lattice $L$ such that $E \cong \mathbb{C}/L$ without actually computing $L$.

**2.1.2 Group operation.** In $\mathbb{C}/L$, the group operation is simply addition (in $\mathbb{C}$) modulo $L$, but what we need is the group operation in $E$ since this is the group we will be working with.

Because $E$ is in bijection with $\mathbb{C}/L$, we will *add* points on the elliptic curve. This addition operation will have neutral element $P_\infty$ since $P_\infty$ is the image of 0 (the neutral element of the addition in $\mathbb{C}/L$) under the bijection, i.e. $P + P_\infty = P$ for every point $P$ on $E$.

For all other pairs of points on the elliptic curve, the group operation is defined using the chord-and-tangent method. Given two points to add, we draw the line between those two points (the tangent line to the curve if we are adding a point to itself) and we find the third point of intersection between the line and the curve (since $E$ is defined by an equation of degree 3, in general a line will intersect the curve in 3 points). Note that for vertical lines, the third point is in fact the point at infinity (to see this correctly, we would have to look at the curve in the projective plane, where $P_\infty$ becomes the "point" $(0:1:0)$). The sum of the two points is then defined as the reflection across the $x$-axis of that third point (for $P_\infty$, the reflection has no effect).

There are two situations in which the chord-and-tangent addition will give $P_\infty$:

- $(x, y) + (x, -y) = P_\infty$
- $2(x, 0) = P_\infty$

(we could also write $2(x, 0) = (x, 0) + (x, -0)$ and reduce to only one case, but the distinction is more obvious from the general formula). From now on, given a point $P = (x, y)$, we will denote the point $(x, -y)$ by $-P$ since it is the additive inverse of $P$.

For the general situation, we let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ (with $P_2 \neq -P_1$), and we compute $P_1 + P_2 = P_3 = (x_3, y_3)$ with

$$\begin{aligned} x_3 &= -x_1 - x_2 + \lambda^2 \\ y_3 &= -y_1 - \lambda(x_3 - x_1), \end{aligned}$$

where $\lambda$ is the slope of the line between the two points (the tangent if both points are the same), i.e.

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 \\[2ex] \dfrac{3{x_1}^2 + a}{2y_1} & \text{if } x_1 = x_2 \end{cases}.$$

We easily see that the addition giving $P_\infty$ are those for which the computation of $\lambda$ would require a division by zero (i.e. we are looking for the third point on a vertical line).

One could verify (although it is too long to be included here) that given $P_1 = \left(\mathcal{P}(z_1), \frac{1}{2}\mathcal{P}'(z_1)\right)$ and $P_2 = \left(\mathcal{P}(z_2), \frac{1}{2}\mathcal{P}'(z_2)\right)$, then $P_3 = P_1 + P_2$ computed using the cord-and-tangent method satisfies $P_3 = \left(\mathcal{P}(z_1 + z_2), \frac{1}{2}\mathcal{P}'(z_1 + z_2)\right)$, so the bijection between $\mathbb{C}/L$ and $E$ is in fact a group isomorphism.

**2.1.3 $n$-torsion groups.** The $n$-torsion group of an elliptic curve $E$, denoted $E[n]$, is the set of points $P$ such that $nP = P_\infty$. In $\mathbb{C}/L$, these points correspond to values of $z$ for which $nz \in L$, i.e.

$$\begin{aligned} E[n] &\cong \left\{ r\frac{\omega_1}{n} + s\frac{\omega_2}{n} \middle| r, s \in \mathbb{Z}/n\mathbb{Z} \right\} \\ &\cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \end{aligned} \tag{2.2}$$

**Lemma 2.4** *If* $\gcd(m, n) = 1$, *then* $E[mn] \cong E[m] \times E[n]$.

**Proof** This is direct from (2.2) since

$$\mathbb{Z}/(mn)\mathbb{Z} \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$$

when $\gcd(m, n) = 1$. □

Although it is very easy to represents the points of $E[n]$ in $\mathbb{C}/L$, finding a characterization for the points on the curve is more complicated and will be done in Section 2.2. For now, we only use (2.2) to obtain a result on endomorphisms of the curve:

**Proposition 2.5** *If* $\sigma \in End(E)$ *(i.e.* $\sigma$ *is an endomorphism of the curve* $E$*), then* $\sigma$ *acts as a* $2 \times 2$ *matrix on* $E[n]$ *and its characteristic polynomial is of the form*

$$\sigma^2 + r\sigma + s = 0$$

*for some integers* $r$ *and* $s$.

From now on, we will not distinguish between the restriction of $\sigma$ on $E[n]$ and the $2 \times 2$ matrix $\tilde{\sigma}$ over $\mathbb{Z}/n\mathbb{Z}$ corresponding to this map.

**2.1.4 Elliptic curves over $\mathbb{F}_q$.** All the theory that we covered so far works in much the same way when applied to finite fields, although a correct proof of all the results would require some extra care.

We say that an elliptic curve $E$ is defined over a field $\mathbb{F}_q$ of characteristic $p > 3$ if it is given by an equation of the form

$$Y^2 = X^3 + aX + b$$

where $a$ and $b$ are elements of $\mathbb{F}_q$ and $27b^2 + 4a^3 \neq 0$ (from Lemma 2.3). Note that $E$ can sometimes be defined over a subfield of $\mathbb{F}_q$, and can always be seen as defined over any extension of $\mathbb{F}_q$, in particular over $\overline{\mathbb{F}}_q$. Although our ultimate goal will be to compute the number of points of $E$ which have their coordinates in $\mathbb{F}_q$ (we write this group as $E(\mathbb{F}_q)$), most of our work will involve points whose coordinates are only defined in an extension of $\mathbb{F}_q$. We will usually denote by $E$ or $E(\overline{\mathbb{F}}_q)$ the group of all points $(x, y)$ satisfying the equation of $E$ with coordinates in **any** field extension of $\mathbb{F}_q$, to which we add the point $P_\infty$.

The group operation described in Section 2.1.2 is easily adapted to $E(\mathbb{F}_q)$ or $E(\overline{\mathbb{F}}_q)$ and the $n$ torsion points (Section 2.1.3) become the set of points $P$ in $E(\overline{\mathbb{F}}_q)$ such that $nP = P_\infty$.

This is where we begin to see major differences between elliptic curves over $\mathbb{C}$ and elliptic curves over finite fields. As long as $n$ is coprime with $p$, we still have $E[n] \cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$, but this is no longer true if $p$ divides $n$ (as could be expected since $n$ torsion points come from dividing the points of the lattice by $n$, which is more problematic in characteristic $p$ dividing $n$). In the case of $E[p]$, two situations can happen:

- $E[p] = \{P_\infty\}$: these curves are called supersingular and are said to have $p$-rank zero. Supersingular elliptic curves are relatively special, and they are weaker from a cryptographic perspective, so we will not be interested in computing the order of these curves (where a trial-and-error approach would be more efficient since there is a very limited number of choices for the group order of supersingular curves over $\mathbb{F}_q$).
- $E[p] \cong \mathbb{Z}/p\mathbb{Z}$: these curves are called ordinary elliptic curves, and are said to have $p$-rank one.

For completeness, note that $E[p^r]$ is $\{P_\infty\}$ for supersingular curves, and isomorphic to $\mathbb{Z}/p^r\mathbb{Z}$ for ordinary curves, and that Proposition 2.4 can be used for all other multiples of $p$.

Because of this change in the form of $E[n]$ when $p$ divides $n$, the remainder of this text will assume that $n < p$ whenever $n$-torsion points are used. Since we are more interested in prime fields, i.e. fields $\mathbb{F}_q$ with $p = q$, the restriction $n < p$ will be almost trivially true.

Finally, Proposition 2.5 can be read almost as-is, with some technical issues coming up for $p$-torsion points. As we just said, the reader can safely ignore those technical issues since we will always assume that $n < p$.

**2.1.5 Frobenius.** For any extension (including $\overline{\mathbb{F}}_q$) of the finite field $\mathbb{F}_q$ we have an automorphism $\sigma$, called the Frobenius, which takes any field element to the $q$-th power. The field $\mathbb{F}_q$ can be seen as the largest subfield of $\overline{\mathbb{F}}_q$ which is fixed under $\sigma$. The Frobenius map can be extended as a map from points on $E$ as follows:

**Definition 2.6** Given a point $P = (x, y) \in E(\overline{\mathbb{F}}_q)$, we define $\phi(P)$ as the point $(\sigma(x), \sigma(y)) = (x^q, y^q)$, and for $P = P_\infty$ we let $\phi(P_\infty) = P_\infty$. This map is called the Frobenius (on $E$).

**Proposition 2.7** *Let $E$ be an elliptic curve defined over $\mathbb{F}_q$. Then the Frobenius (on $E$) defines an isomorphism $\phi : E \to E$ and $\phi - 1$ has kernel $E(\mathbb{F}_q)$.*

**Proof** Let $P = (x_0, y_0) \in E(\overline{\mathbb{F}}_q)$, i.e. $x_0, y_0 \in \overline{\mathbb{F}}_q$ and satisfying $y_0^2 = x_0^3 + ax_0 + b$. Then $\phi(P) = (x_0^q, y_0^q)$ is also a point in $E(\overline{\mathbb{F}}_q)$ since $x_0^q$ and $y_0^q$ are in $\overline{\mathbb{F}}_q$ and

$$\begin{aligned}(y_0^q)^2 &= (y_0^2)^q \\ &= (x_0^3 + ax_0 + b)^q \\ &= (x_0^3)^q + a^q x_0^q + b^q \\ &= (x_0^q)^3 + a \cdot x_0^q + b \ .\end{aligned}$$

Since $\phi$ leaves $P_\infty$ (the neutral element in the group $E$) fixed, $\phi$ is an endomorphism on $E$. Since $\sigma$ is invertible in $\overline{\mathbb{F}}_q$, $\phi$ must also be invertible (by inverting each coordinate separately), so $\phi$ is an automorphism of $E$. For $P \in E \setminus P_\infty$, $P$ is fixed under $\phi$ if and only if $(x_0, y_0) = (x_0^q, y_0^q)$, i.e. if both $x_0$ and $y_0$ are fixed under $\sigma$, so $P$ is fixed under the Frobenius of $E$ if and only if $P \in E(\mathbb{F}_q)$. □

From Proposition 2.5, the characteristic polynomial of $\phi$ (on $E$) must be of the form

$$\phi^2 + r\phi + s = 0$$

for some integers $r$ and $s$.

Given any point $P \in E(\mathbb{F}_q)$, the order of $P$ must divide $1 + r + s$ (since $P_\infty = \phi^2(P) + r\phi(P) + sP = P + rP + sP = (1 + r + s)P$). The relation between

the Frobenius and the number of points on $E(\mathbb{F}_q)$ is even stronger than the one hinted at in this argument, as can be seen in the Hasse–Weil Theorem:

**Theorem 2.8** (Hasse–Weil) *Given an elliptic curve $E$ defined over $\mathbb{F}_q$, the Frobenius endomorphism $\phi$ on $E$ satisfies a quadratic equation of the form*

$$\phi^2 - t\phi + q = 0$$

*with $|t| \leq 2\sqrt{q}$ and the order of the group is*

$$|E(\mathbb{F}_q)| = q + 1 - t \ .$$

Although the proof of this theorem is beyond the scope of this text, the result itself is essential to our purpose.

## 2.2 Division polynomials

In this section, we introduce the *division polynomials*, and give some of their properties. These division polynomials can be used to describe the $n$-torsion points and they are the building blocks of Schoof's point counting algorithm.

We define the division polynomial $\Psi_n(X, Y)$ using:

$$\Psi_n(X,Y)^2 = n^2 \prod_{P \in E[n] \setminus P_\infty} (X - \mathcal{X}(P)) \ .$$

Clearly $\Psi_n^2$ is a polynomial in $\overline{\mathbb{F}}_q[X]$, and since the Frobenius is an isomorphism on $E[n]$ (since $n\phi(P) = \phi(nP) = \phi(P_\infty) = P_\infty$), $\Psi_n^2$ is fixed under $\phi$, so it is in fact a polynomial in $\mathbb{F}_q[X]$. To avoid any side issues due to the $p$-torsion groups, we will restrict to values of $n$ smaller than the characteristic of $\mathbb{F}_q$ (i.e. $n < q$ in the setting of prime fields).

Before looking at $\Psi_n$ for a general $n$, we will look at the 2-torsion group. The points in $E[2] \setminus P_\infty$ are the affine points satisfying $2P = P_\infty$, and from Section 2.1.2 we know that those are exactly the points with $y$ coordinate zero. If the $y$ coordinate of a point is zero, then its $x$ coordinate must be one of the (three) roots of $X^3 + aX + b = 0$. We then have

$$\Psi_2(X,Y)^2 = 4 \prod_{P \in E[2] \setminus P_\infty} (X - \mathcal{X}(P)) = 4 \prod_{P \in E, \mathcal{Y}(P)=0} (X - \mathcal{X}(P)) = Y^2$$

since $E$ is defined by the equation $Y^2 = X^3 + aX + b$.

We can now show that $\Psi_n$ is indeed a polynomial in $\mathbb{F}_q[X, Y]$:

- If $n < p$ is odd:
  If $P$ is in $E[n] \setminus P_\infty$, then $-P$ is also in $E[n] \setminus P_\infty$ (since $n(-P) = -nP = -P_\infty = P_\infty$) and $-P \neq P$ (otherwise we would have $2P = P+P = P-P = P_\infty$, which is impossible in $E[n] \setminus P_\infty$). Since $P$ and $-P$ have the same $x$ coordinate, the pair $P$, $-P$ contributes a factor $(X - \mathcal{X}(P))^2$ to $\Psi_n(X,Y)^2$.

  All the points in $E[n] \setminus P_\infty$ are in exactly one pair $P$, $-P$ (there are $\frac{n^2-1}{2}$ such pairs), so $\Psi_n^2$ is a perfect square and we find that $\Psi_n$ is well defined as a polynomial in $\mathbb{F}_q[X]$.
- If $n < p$ is even:
  If $P$ is in $E[n] \setminus E[2]$, we have the same argument as above, but with $n^2 - 4$ points ($\frac{n^2-4}{2}$ pairs), and we get a square factor with a well defined square root in $\mathbb{F}_q[X]$.

The remaining factor of $\Psi_n^2$ comes from the points in $E[2] \setminus P_\infty$, which contribute a factor $Y^2$, so $\Psi_n^2$ is a perfect square and we find that $\Psi_n$ is well defined as a polynomial in $Y\mathbb{F}_q[X]$.

**Definition 2.9** For $n < p$ odd, we set $f_n(X) = \Psi_n(X,Y)$, of degree $(n^2-1)/2$, and for $n < q$ even, we set $f_n(X) = \Psi_n(X,Y)/Y$, of degree $(n^2-4)/2$. The leading coefficient of $f_n(X)$ is $n$.

The polynomials $f_n(X) \in \mathbb{F}_q[X]$ are used for computations instead of the polynomials $\Psi_n(X,Y)$ which can be in $\mathbb{F}_q[X]$ ($n$ even) or $Y\mathbb{F}_q[X]$ ($n$ odd). To simplify some computations, we extend this definition for $n < 0$ via $f_{-n}(X) = -f_n(X)$. To make some equations easier to read, we will write $\Psi_n(P)$ for $\Psi_n(\mathcal{X}(P), \mathcal{Y}(P))$, $\Psi_n$ for $\Psi_n(X,Y)$ and $f_n$ for $f_n(X)$. In order to deal with poles and zeros at $P_\infty$, we will also need the function (on $\mathbb{C}/L$) $\Psi_n(z) = \Psi_n(\mathcal{X}(z), \mathcal{Y}(z))$.

The following proposition follows directly from our definitions of $\Psi_n(X,Y)$ and $f_n(X)$. It will allow us to work on $E[n] \setminus P_\infty$ without having to actually compute the $n$-torsion points.

**Proposition 2.10** *For $0 < n < p$, we have:*

1. *$\Psi_n(X,Y)$ and $f_n(X)$ do not contain any non-constant square factors;*
2. *$P \in E[n] \setminus P_\infty$ if and only if $\Psi_n(P) = 0$;*
3. *$P \in E[n] \setminus E[2]$ if and only if $f_n(\mathcal{X}(P)) = 0$; and*
4. *$P \in E[2] \setminus P_\infty$ if and only if $\mathcal{Y}(P) = 0$.*

**2.2.1 Properties.** The polynomials $\Psi_n(X,Y)$ satisfy a number a nice properties which will make it possible to compute them efficiently (Corollary 2.15 and Proposition 2.17), as well as describing the $x$ and $y$ coordinates of the point $mP$ in terms of the coordinates of $P$ (Corollary 2.13 and Proposition 2.16).

The reader should be warned that the proofs of Propositions 2.12 and 2.16 are quite technical (although very similar to each other), and will make use of the following lemma:

**Lemma 2.11** *For $n < p$, $\Psi_n(z) = \frac{n(-1)^{n+1}}{z^{n^2-1}}(1 + O(z^4))$*

**Proof** We recall that $\mathcal{X}(z) = \frac{1}{z^2} + O(z^2) = \frac{1}{z^2}(1 + O(z^4))$ and $\mathcal{Y}(z) = \frac{-1}{z^3} + O(z) = \frac{-1}{z^3}(1 + O(z^4))$. When we substitute $\mathcal{X}(z)$ and $\mathcal{Y}(z)$ in $\Psi_n(X,Y)$, we only need to worry about the term of highest (total) degree (in $X$ and $Y$), i.e. $nX^{(n^2-1)/2}$ if $n$ is odd and $YX^{(n^2-4)/2}$ if $n$ is even (using Definition 2.9). The identity can then be completed with an easy computation. □

**Proposition 2.12** *For every $m \neq 0$, $n \neq 0$, $n \neq m$ ($n+m < p$) and $P \in E(\overline{\mathbb{F}_q})$,*

$$\mathcal{X}(mP) - \mathcal{X}(nP) = -\frac{\Psi_{m+n}(P)\Psi_{m-n}(P)}{\Psi_m(P)^2\Psi_n(P)^2}$$

**Proof** To show this equality, we will work with the functions

$$A(P) = \mathcal{X}(mP) - \mathcal{X}(nP)$$

and

$$B(P) = -\frac{\Psi_{m+n}(P)\Psi_{m-n}(P)}{\Psi_m(P)^2\Psi_n(P)^2} \quad ,$$

from which we get two elliptic functions $A(z)$ and $B(z)$ via the map relating $z$ to $(\mathcal{X}(z), \mathcal{Y}(z))$ (the isomorphism between $\mathbb{C}/L$ and $E$). To prove they are equal, it

will therefore be sufficient to show that they have the same poles and zeros (with multiplicity) and to match one of their (non-zero) coefficients.

To do this, we will have to consider the zeros and poles in $E[d]$ with $d = \gcd(m, n)$ separately from the others. Our approach will be to use the zeros and poles of $A(P)$ in $E \setminus E[d]$ to argue the form of $B(P)$ in terms of the $\Psi_i(P)$ (assuming the points in $E[d]$ behave as we would like), and then to show that the two functions are also a match in $E[d]$.

Poles in $E \setminus E[d]$: The poles of $A(P)$ are the points in $E[m] \cup E[n]$ since $\mathcal{X}(mP) + \mathcal{X}(nP) = \infty$ if and only if $mP = P_\infty$ or $nP = P_\infty$. If $P \in E[m] \setminus E[d]$, then $\Psi_m(P) = 0$ while the other factors of $B(P)$ are nonzero. Since multiplication by $m$ maps $z \in E[m] \setminus E[d]$ to $L$ which is a double pole of $\mathcal{X}(z)$, every $P \in E[m] \setminus E[d]$ must be a double pole of $B(z)$, so the denominator should contain a factor of ${\Psi_m}^2$. By the same argument on $E[n] \setminus E[d]$, the denominator should also contain a factor of ${\Psi_n}^2$. Since there are no other poles outside of $E[d]$, we find $(\Psi_m \Psi_m)^2$ for the denominator of $B(z)$.

Zeros in $E \setminus E[d]$: In $E \setminus (E[m] \cup E[n])$, the zeros of $A(P)$ are the points $P$ such that $\mathcal{X}(mP) = \mathcal{X}(nP)$, i.e. such that $mP = \pm nP$. If $P$ is a zero of $A(P)$ in $E \setminus E[d]$ ($mP = P_\infty = nP$ if and only if $P \in E[d]$), then it is in one of two cases: $mP + nP = (m+n)P = P_\infty$ ($P$ is a root of $\Psi_{m+n}$) or $mP - nP = (m-n)P = P_\infty$ ($P$ is a root of $\Psi_{m-n}$).

To find the multiplicity of the zeros, we look at the derivative of $A(z)$:

$$A'(z) = m\mathcal{X}'(mz) - n\mathcal{X}'(nz) = 2(m\mathcal{Y}(mz) - n\mathcal{Y}(nz)) \ .$$

But If $P$ is a zero of $A(P)$ in $E \setminus E[d]$, then $mP = \pm nP$, so $\mathcal{Y}(mP) = \pm\mathcal{Y}(nP)$ and we are looking at the identity

$$2(m \pm n)\mathcal{Y}(mP) \ ,$$

which is zero if and only if $\mathcal{Y}(mP) = 0$, so if and only if $mP = nP$ is a point of order 2, i.e. $P \in E[2m] \cap E[2n] = E[2d]$. The zeros of $A(P)$ outside of $E[2d]$ will therefore have multiplicity one. For $P \in E[2d] \setminus E[d]$, we look at the second derivative:

$$A''(z) = m^2\mathcal{X}''(mz) - n^2\mathcal{X}''(nz) = m^2(6\mathcal{X}(mz)^2 + 2a) - n^2(6\mathcal{X}(nz)^2 + 2a)$$

If $P \in E[2d]$, then once again $\mathcal{X}(mP) = \mathcal{X}(nP)$, so we are looking at the identity

$$2(m^2 - n^2)(3\mathcal{X}(mP)^2 + a) \ ,$$

which cannot be zero by Lemma 2.3 since $\mathcal{X}(mP)$ is already a zero of $\mathcal{X}(mP)^3 + a\mathcal{X}(mP) + b$ (as $\mathcal{Y}(mP) = 0$). If $P$ is a zero in $E[2d] \setminus E[d]$ it must therefore have multiplicity two.

If both $(m+n)P$ and $(m-n)P$ are equal to $P_\infty$, then $2mP = P_\infty = 2nP$, so $P \in E[2d] \setminus E[d]$ (note that $m$ and $n$ must both be odd for this to be possible). In this case, $P$ is a double zero and it is a zero (with multiplicity one) of both $\Psi_{m+n}$ and $\Psi_{m-n}$. Outside of $E[2d]$, $P \in E \setminus E[2d]$ is a zero of $A(P)$ if and only if it is a zero of either $\Psi_{m+n}$ or $\Psi_{m-n}$, and the two sets of zeros are disjoint (and have multiplicity one) outside of $E[d]$. For the zeros outside of $E[d]$, the numerator of $B(P)$ should then be a constant multiple of $\Psi_{m+n}\Psi_{m-n}$.

Poles and zeros in $E[d]$: For every $P \in E[d] \setminus P_\infty$, $P$ is a double pole of $\mathcal{X}(mP)$ and $\mathcal{X}(nP)$, so $P$ is a double pole of $A$. Since $d$ divides both $m$ and $n$, $P$ is a (single) zero of $\Psi_m$, $\Psi_n$, $\Psi_{m+n}$, and $\Psi_{m-n}$, so $P$ is a double pole of $B$ (after cancellations).

All that remains is to verify that $P_\infty$ is a pole of order two of both functions and that at least one of their coefficients is the same.

We now look at the Laurent series of $A(z)$:

$$A(z) = \mathcal{X}(mz) - \mathcal{X}(nz) = \left(\frac{1}{m^2} - \frac{1}{n^2}\right)\frac{1}{z^2} + O(z^2) \ .$$

Looking at the Laurent series of $B(z)$, we find

$$\begin{aligned}
B(z) &= -\frac{\Psi_{m+n}(z)\Psi_{m-n}(z)}{\Psi_m(z)^2\Psi_n(z)^2} \\
&= -\frac{\left(\frac{(m+n)(-1)^{m+n+1}}{z^{(m+n)^2-1}}(1+O(z^4))\right)\cdot\left(\frac{(m-n)(-1)^{m-n+1}}{z^{(m-n)^2-1}}(1+O(z^4))\right)}{\left(\frac{m(-1)^{m+1}}{z^{m^2-1}}(1+O(z^4))\right)^2\cdot\left(\frac{n(-1)^{n+1}}{z^{n^2-1}}(1+O(z^4))\right)^2} \\
&= -\frac{(m+n)(m-n)}{m^2n^2(-1)^{2n+2}}\frac{z^{2(m^2-1)+2(n^2-1)}}{z^{(m^2+2mn+n^2-1)+(m^2-2mn+n^2-1)}}(1+O(z^4)) \\
&= \left(\frac{1}{m^2} - \frac{1}{n^2}\right)\frac{1}{z^2} + O(z^2) \ ,
\end{aligned}$$

so $B(z)$ as a double pole at $z = 0$, completing the match for all the poles and zeros. Since the coefficient of $z^{-2}$ is the same as in the Laurent series of $A(z)$, the two functions are equal. □

**Corollary 2.13** *For every $m \in \mathbb{N}$, $m + 1 < p$,*

$$\mathcal{X}(mP) = \mathcal{X}(P) - \frac{\Psi_{m+1}(P)\Psi_{m-1}(P)}{\Psi_m(P)^2} \ .$$

**Proof** If $m$ is not 0 or 1, we can use Proposition 2.12 with $n = 1$ and the result is straightforward. For $m = 0$, we have

$$\begin{aligned}
\mathcal{X}(P_\infty) = \mathcal{X}(0P) &= \mathcal{X}(P) - \frac{\Psi_1(P)\Psi_{-1}(P)}{\Psi_0(P)^2} \\
&= \mathcal{X}(P) - \frac{-1}{0^2} \\
&= \infty
\end{aligned}$$

(where the $\infty$ corresponds to a pole of multiplicity 2), and for $m = 1$ we get

$$\begin{aligned}
\mathcal{X}(P) = \mathcal{X}(1P) &= \mathcal{X}(P) - \frac{\Psi_{1+1}\Psi_{1-1}}{\Psi_1^2} \\
&= \mathcal{X}(P) - \frac{0}{1} \\
&= \mathcal{X}(P) \ .
\end{aligned}$$

□

**Proposition 2.14** *For every $m > n$, $m + n < p$,*

$$\Psi_{m+n}\Psi_{m-n} = \Psi_{m+1}\Psi_{m-1}\Psi_n^2 - \Psi_{n+1}\Psi_{n-1}\Psi_m^2$$

**Proof** From Proposition 2.12, we have

$$\mathcal{X}(mP) - \mathcal{X}(nP) = -\frac{\Psi_{m+n}(P)\Psi_{m-n}(P)}{\Psi_m(P)^2\Psi_n(P)^2}$$

and from Corollary 2.13, we get

$$\mathcal{X}(P) - \mathcal{X}(nP) = \frac{\Psi_{n+1}(P)\Psi_{n-1}(P)}{\Psi_n(P)^2}$$

and

$$\mathcal{X}(P) - \mathcal{X}(mP) = \frac{\Psi_{m+1}(P)\Psi_{m-1}(P)}{\Psi_m(P)^2} \ .$$

Since $(\mathcal{X}(P) - \mathcal{X}(nP)) - (\mathcal{X}(P) - \mathcal{X}(mP)) = \mathcal{X}(mP) - \mathcal{X}(nP)$, we find

$$\frac{\Psi_{n+1}(P)\Psi_{n-1}(P)}{\Psi_n(P)^2} - \frac{\Psi_{m+1}(P)\Psi_{m-1}(P)}{\Psi_m(P)^2} = -\frac{\Psi_{m+n}(P)\Psi_{m-n}(P)}{\Psi_m(P)^2\Psi_n(P)^2}$$

for every $P \in E(\overline{\mathbb{F}_q})$, from which we get the desired equation. □

**Corollary 2.15** *For every* $2n+1 < p$,

$$\begin{aligned}\Psi_{2n+1} &= \Psi_1\Psi_{2n+1} \\ &= \Psi_{n+2}\Psi_n^3 - \Psi_{n+1}^3\Psi_{n-1} \\ 2Y\Psi_{2n} &= \Psi_2\Psi_{2n} \\ &= \Psi_n(\Psi_{n+2}\Psi_{n-1}^2 - \Psi_{n-2}\Psi_{n+1}^2)\end{aligned}$$

**Proof** We apply Proposition 2.14 with $n+1 > n$ $(m > n)$ to get $\Psi_{2n+1}$ and $n+1 > n-1$ to get $2Y\Psi_{2n}$. □

**Proposition 2.16** *For every* $n \in \mathbb{N}$, $n+2 < p$,

$$\mathcal{Y}(nP) = \frac{\Psi_{n+2}(P)\Psi_{n-1}(P)^2 - \Psi_{n-2}(P)\Psi_{n+1}(P)^2}{4\mathcal{Y}(P)\Psi_n(P)^3}$$

**Proof** We first note that

$$\mathcal{Y}(nP) = \infty \iff nP = P_\infty$$

and

$$\mathcal{Y}(nP) = 0 \iff (nP \neq P_\infty \text{ and } 2nP = P_\infty).$$

Assuming that $P \neq P_\infty$, we have $\Psi_n(P) = 0 \iff nP = P_\infty$, so

$$\mathcal{Y}(nP) = \infty \iff \Psi_n(P) = 0 \ .$$

By the same argument, $\Psi_{2n}(P) = 0 \iff 2nP = P_\infty$ is combined with Corollary 2.15,

$$2\mathcal{Y}(P)\Psi_{2n}(P) = \Psi_n(P)(\Psi_{n+2}(P)\Psi_{n-1}(P)^2 - \Psi_{n-2}(P)\Psi_{n+1}(P)^2) \ ,$$

to give

$$\mathcal{Y}(nP) = 0 \iff (\Psi_{n+2}(P)\Psi_{n-1}(P)^2 - \Psi_{n-2}(P)\Psi_{n+1}(P)^2)/2\mathcal{Y}(P) = 0.$$

The remainder of the proof follows the same ideas used to prove Proposition 2.12.
□

**Proposition 2.17** *The first (initial) values of* $f_n(X)$ *are:*

1. $f_0 = 0$
2. $f_1 = 1$
3. $f_2 = 2$
4. $f_3(X) = 3X^4 + 6aX^2 + 12bX - a^2$
5. $f_4(X) = 4X^6 + 20aX^4 + 80bX^3 - 20a^2X^2 - 16abX - 32b^2 - 4a^3$

**Proof** The different $f_n(X)$ are found by constructing $\Psi_n(X,Y)$:

1. By definition, ${\Psi_0}^2 = 0$ (there are no 0-torsion points).
2. The only 1-torsion point is $P_\infty$, so ${\Psi_1}^2 = 1$.
3. By definition, ${\Psi_2}^2 = 2^2Y^2$, so $\Psi_2 = 2Y$ and $f_2 = \Psi_2/Y = 2$.
4. From Corollary 2.13, we have

$$\begin{aligned}\mathcal{X}(2P) &= \mathcal{X}(P) - \frac{\Psi_3(P)\Psi_1(P)}{\Psi_2(P)^2} \\ &= \mathcal{X}(P) - \frac{\Psi_3(P)}{4\mathcal{Y}(P)^2}\ . \end{aligned} \tag{2.3}$$

Using the point doubling formula (Section 2.1.2), allowing $\lambda = \infty$ (i.e. using the same formula to double points of order 2), we have:

$$\begin{aligned}\mathcal{X}(2P) &= -2\mathcal{X}(P) + \left(\frac{d\mathcal{Y}}{d\mathcal{X}}(P)\right)^2 \\ &= -2\mathcal{X}(P) + \left(\frac{\mathcal{Y}'(P)}{\mathcal{X}'(P)}\right)^2 \\ &= -2\mathcal{X}(P) + \left(\frac{\frac{1}{2}\mathcal{X}''(P)}{2\mathcal{Y}(P)}\right)^2 \\ &= -2\mathcal{X}(P) + \frac{(3\mathcal{X}(P)^2+a)^2}{4\mathcal{Y}(P)^2} && (2.4) \\ &= \mathcal{X}(P) - \frac{3\mathcal{X}(P)\cdot 4\mathcal{Y}(p)^2 - (3\mathcal{X}(P)^2+a)^2}{4\mathcal{Y}(P)^2} \\ &= \mathcal{X}(P) - \frac{12\mathcal{X}(P)(\mathcal{X}(P)^3+a\mathcal{X}(P)+b)-(9\mathcal{X}(P)^4+6\mathcal{X}(P)^2+a^2)}{4\mathcal{Y}(P)^2} \\ &= \mathcal{X}(P) - \frac{3\mathcal{X}(P)^4+6a\mathcal{X}(P)^2+12b\mathcal{X}(P)-a^2}{4\mathcal{Y}(P)^2} && (2.5)\end{aligned}$$

(recall that $\mathcal{Y}(P) = \frac{1}{2}\mathcal{X}'(P)$ and $\mathcal{X}''(P) = 6\mathcal{X}(P)^2 + 2a$). Combining Equations 2.5 and 2.3, we can solve for $\Psi_3 = f_3$.

5. Let $u, v \in E$ be such that $u \neq \pm v$, then the addition formula gives us:

$$\begin{aligned}\mathcal{X}(u+v) &= -\mathcal{X}(u) - \mathcal{X}(v) + \left(\frac{\mathcal{Y}(u)-\mathcal{Y}(v)}{\mathcal{X}(u)-\mathcal{X}(v)}\right)^2 \\ \mathcal{X}(u-v) &= -\mathcal{X}(u) - \mathcal{X}(v) + \left(\frac{\mathcal{Y}(u)+\mathcal{Y}(v)}{\mathcal{X}(u)-\mathcal{X}(v)}\right)^2\ .\end{aligned}$$

Assuming that $2P \neq \pm P$ (i.e. that $P \notin E[3]$, which is not a problem since $\Psi_4$ comes from the elements of $E[4] \setminus P_\infty$), we let $u = 2P$ and $v = P$, and

we find:

$$
\begin{aligned}
\mathcal{X}(3P) - \mathcal{X}(P) &= \mathcal{X}(2P+P) - \mathcal{X}(2P-P) \\
&= \left(-\mathcal{X}(2P) - \mathcal{X}(P) + \left(\frac{\mathcal{Y}(2P) - \mathcal{Y}(P)}{\mathcal{X}(2P) - \mathcal{X}(P)}\right)^2\right) \\
&\quad - \left(-\mathcal{X}(2P) - \mathcal{X}(P) + \left(\frac{\mathcal{Y}(2P) + \mathcal{Y}(P)}{\mathcal{X}(2P) - \mathcal{X}(P)}\right)^2\right) \\
&= \frac{(\mathcal{Y}(2P) - \mathcal{Y}(P))^2}{(\mathcal{X}(2P) - \mathcal{X}(P))^2} - \frac{(\mathcal{Y}(2P) + \mathcal{Y}(P))^2}{(\mathcal{X}(2P) - \mathcal{X}(P))^2} \\
&= \frac{-4\mathcal{Y}(2P)\mathcal{Y}(P)}{(\mathcal{X}(2P) - \mathcal{X}(P))^2} \\
&= \frac{-4\mathcal{Y}(2P)\mathcal{Y}(P)}{\left(-\frac{\Psi_3(P)\Psi_1(P)}{\Psi_2(P)^2}\right)^2} \\
&= -\frac{\mathcal{Y}(2P) \cdot 64\mathcal{Y}(P)^5}{\Psi_3(P)^2}
\end{aligned}
$$

From Proposition 2.12, we have

$$
\begin{aligned}
\mathcal{X}(3P) - \mathcal{X}(P) &= -\frac{\Psi_4(P)\Psi_2(P)}{\Psi_3(P)^2} \\
&= -\frac{\Psi_4(P) \cdot 2\mathcal{Y}(P)}{\Psi_3(P)^2}
\end{aligned}
$$

Which gives us

$$
\Psi_4(P) = 4\mathcal{Y}(2P) \cdot 8\mathcal{Y}(P)^4 \tag{2.6}
$$

From (2.4), we have

$$
\begin{aligned}
\mathcal{X}(2P) &= \frac{-2\mathcal{X}(P) \cdot 4(\mathcal{X}(P)^3 + a\mathcal{X}(P) + b) + (3\mathcal{X}(P)^2 + a)^2}{4\mathcal{Y}(P)^2} \\
&= \frac{\mathcal{X}(P)^4 - 2a\mathcal{X}(P)^2 - 8b\mathcal{X}(P) + a^2}{4\mathcal{Y}(P)^2} \quad , 
\end{aligned} \tag{2.7}
$$

and since $(\mathcal{X}(2P))' = 2 \cdot 2\mathcal{Y}(2P)$, we obtain

$$
\begin{aligned}
4\mathcal{Y}(P) &= (\mathcal{X}(2P))' \\
&= \left(\frac{\mathcal{X}(P)^4 - 2a\mathcal{X}(P)^2 - 8b\mathcal{X}(P) + a^2}{4\mathcal{Y}(P)^2}\right)' \\
&= \frac{\left(\mathcal{X}(P)^4 - 2a\mathcal{X}(P)^2 - 8b\mathcal{X}(P) + a^2\right)'}{4\mathcal{Y}(P)^2} \\
&\quad - \frac{\left(\mathcal{X}(P)^4 - 2a\mathcal{X}(P)^2 - 8b\mathcal{X}(P) + a^2\right)\mathcal{Y}'(P)}{2\mathcal{Y}(P)^3} \\
&= \frac{\left(4\mathcal{X}(P)^3 - 4a\mathcal{X}(P) - 8b\right) \cdot 2\mathcal{Y}(P)}{4\mathcal{Y}(P)^2} \\
&\quad - \frac{\left(\mathcal{X}(P)^4 - 2a\mathcal{X}(P)^2 - 8b\mathcal{X}(P) + a^2\right) \cdot \frac{1}{2}\mathcal{X}''(P)}{2\mathcal{Y}(P)^3} \\
&= \frac{\left(4\mathcal{X}(P)^3 - 4a\mathcal{X}(P) - 8b\right)\left(\mathcal{X}(P)^3 + a\mathcal{X}(P) + b\right)}{2\mathcal{Y}(P)^3} \\
&\quad - \frac{\left(\mathcal{X}(P)^4 - 2a\mathcal{X}(P)^2 - 8b\mathcal{X}(P) + a^2\right)\left(3\mathcal{X}(P)^2 + a\right)}{2\mathcal{Y}(P)^3} \\
&= \frac{\left(\mathcal{X}(P)^6 + 5a\mathcal{X}(P)^4 + 20b\mathcal{X}(P)^3 - 5a^2\mathcal{X}(P)^2 - 4ab\mathcal{X}(P) - 8b^2 - a^3\right)}{2\mathcal{Y}(P)^3}
\end{aligned}
$$

Multiplying by $8\mathcal{Y}(P)^4$ gives us $\Psi_4(P)$ for every $P \in E(\overline{\mathbb{F}_q}) \setminus E[3]$, from which we get $\Psi_4(X,Y)$ and $f_4(X) = \Psi_4(X,Y)/Y$.

□

Using Propositions 2.14 and 2.17 and remembering that $f_{-n} = -f_n$, we can now compute $f_n$ for every $n \in \mathbb{Z}$, $-p < n < p$.

## 2.3 Schoof's algorithm

From Theorem 2.8, we know that the Frobenius endomorphism $\phi(x,y) = (x^q, y^q)$ on $E$ satisfies $\phi^2 - t\phi + q = 0$. with $|t| \leq 2\sqrt{q}$ and $|E(\mathbb{F}_q)| = q + 1 - t$.

The idea of Schoof's algorithm is to compute $t_\ell$ for a number of small primes $\ell$ by looking at the effect of the Frobenius on the $\ell$-torsion points. For any point $P$ (or $\phi(P)$) in $E[\ell] \setminus P_\infty$, $\ell P = P_\infty$ so the Frobenius must satisfy an equation of the form

$$\phi^2 - t_\ell \phi + k = 0 \tag{2.8}$$

for every point in $e[\ell] \setminus P_\infty$, with $k \equiv q \bmod \ell$ and $t_\ell \in \{0, 1, \ldots, \ell - 1\}$.

The algorithm takes advantage of the division polynomials and their properties to compute $t_\ell$ for values of $\ell$ less than a upper bound $L$ satisfying

$$\prod_{\substack{\ell \text{ prime} \\ \ell < L}} \ell > 4\sqrt{q}$$

The value of $t$ can then be computed using the Chinese Remainder Theorem.

**2.3.1 Special case: $\ell = 2$.** Since $f_2(X) = 2$, we cannot get any information on $t \bmod 2$ by working modulo $f_2(X)$. However, we have $q + 1 - t \equiv t \bmod 2$ (since $q$ is odd) so we only need to determine if $|E(\mathbb{F}_q)|$ is even or odd. By Lagrange's Theorem, $|E(\mathbb{F}_q)|$ is even if and only if it contains a point of order 2 (other than $P_\infty$). Since the only points $P$ on $E$ for which $2P = P_\infty$ are of the form $(x, 0)$ (see Section 2.1.2), this is equivalent to asking if $X^3 + aX + b$ has a zero in $\mathbb{F}_q$, and the test for $t_2$ becomes:

**Test 2.18** *To compute* $t \bmod 2$*, determine if*

$$\gcd\left(X^q - X, X^3 + aX + b\right) \begin{cases} = 1 & (t_2 = 1) \\ \neq 1 & (t_2 = 0) \end{cases}.$$

**2.3.2 Prime $\ell \neq 2$.** To find $t \bmod \ell$, $\ell > 2$, we will look for a value of $\tau \in \{1, 2, \ldots, \ell - 1\}$ such that $\tau\phi(P) = \phi^2(P) + kP$ for every $P \in E[\ell] \setminus P_\infty$. Since the general case of the point addition of $\phi^2(P)$ and $kP$ only applies if the $x$ coordinates of the two points to add are distinct, we will distinguish between two cases:

- if $\phi^2(P_0) = \pm kP_0$ for some $P_0 \in E[l] \setminus P_\infty$ (Case 1);
- if $\phi^2(P) \neq \pm kP$ for every $P \in E[l] \setminus P_\infty$ (Case 2).

Given $P_0 = (X_0, Y_0)$, we have

$$\phi^2(P_0) = \left(X_0^{q^2}, Y_0^{q^2}\right)$$

(by the definition of the Frobenius) and $kP_0 =$

$$\left(X_0 - \frac{\Psi_{k+1}(P_0)\Psi_{k-1}(P_0)}{\Psi_k(P_0)^2}, \frac{\Psi_{k+2}(P_0)\Psi_{k-1}(P_0)^2 - \Psi_{k-2}(P_0)\Psi_{k+1}(P_0)^2}{4Y_0\Psi_k(P_0)^3}\right)$$

(by Corollary 2.13 and Proposition 2.16). Asking if $\phi^2(P_0) = \pm kP_0$ for some $P_0 \in E[\ell] \setminus P_\infty$ is equivalent to asking if

$$X_0^{q^2} = X_0 - \frac{\Psi_{k-1}(X_0, Y_0)\Psi_{k+1}(X_0, Y_0)}{\Psi_k^2(X_0, Y_0)} \tag{2.9}$$

for some root $(X_0, Y_0)$ of $\Psi_\ell(X, Y)$ (i.e. for the $x$ and $y$ coordinates of the point $P_0$).

For all $P \in E[\ell] \setminus P_\infty$, $\Psi_k(P) \neq 0$ since $\Psi_k(P) = 0$ would imply that $qP = kP = P_\infty$ (and since $P$ has order $\ell$, a prime, it would force $q$ to be divisible by $\ell$), so $\Psi_k(X, Y)$ is invertible modulo $\Psi_\ell(X, Y)$. We can therefore multiply the function of $X$ and $Y$ corresponding to (2.9) by $\Psi_k^2$ and check whether or not we have an equality for some point of order $\ell$, that is if the polynomial

$$\left(X^{q^2} - X\right)\Psi_k^2(X, Y) + \Psi_{k+1}(X, Y)\Psi_{k-1}(X, Y)$$

has a common root with $\Psi_\ell(X, Y)$.

The first test is therefore:

**Test 2.19** *For* $k \equiv q \bmod \ell$*, determine if*

$$\gcd\left((X^{q^2} - X)\Psi_k^2 + \Psi_{k-1}\Psi_{k+1}, f_\ell\right) \begin{cases} \neq 1 & \textit{(case 1)} \\ = 1 & \textit{(case 2)} \end{cases}.$$

Note: the complete process of the algorithm for a given value of $\ell$ will be described in Algorithm 1.

**2.3.3 Case 1.** Let $P_0 \in E[\ell] \setminus P_\infty$ be such that $\phi^2(P_0) = \pm kP_0$.

As can be expected, case 1 can be divided into two subcases depending on the sign of the equality:

- If $\phi^2(P_0) = -kP_0$, then $\phi^2(P_0) - t_\ell\phi(P_0) + kP_0 = -kP_0 - t_\ell\phi(P_0) + kP_0 = P_\infty$ and we have $t_\ell\phi(P_0) = P_\infty$. Since $\phi$ is an automorphism on $E[\ell]$, $r\phi(P_0) \neq P_\infty$ if $\gcd(r, \ell) = 1$, so $t_\ell$ must be 0.
- On the other hand, if $\phi^2(P_0) = kP_0$, then $\phi^2(P_0) - t_\ell\phi(P_0) + kP_0 = kP_0 - t_\ell\phi(P_0) + kP_0 = P_\infty$, so $(2k - t_\ell\phi)P_0 = P_\infty$. We get that $\phi$ acts as scalar multiplication by $\frac{2k}{t_\ell} \bmod \ell$ ($t_\ell$ must be invertible modulo $\ell$ since $2kP_0 \neq P_\infty$ and $\phi(P_0) \neq P_\infty$), and since $\phi^2(P_0) = kP_0$, we must have $t_\ell^2 \equiv 4k \bmod \ell$. Let $w$ be such that $w^2 \equiv k \bmod \ell$ (such a $w$ exists since $4k$ is a square modulo $\ell$), then either $\phi(P_0) = wP_0$ (and $t_\ell \equiv \frac{2k}{w} \equiv 2w \bmod \ell$) or $\phi(P_0) = -wP_0$ (and $t_\ell \equiv \frac{2k}{-w} \equiv -2w \bmod \ell$).

Furthermore, these two subcases are mutually exclusive.

We could use the $y$ coordinate to distinguish between the two subcases, but it is more efficient to "try" to fit $P_0$ into the subcase $\phi^2(P_0) = kP_0$, and go back to the subcase $\phi^2(P_0) = -kP_0$ if this fails.

If we are in the subcase $\phi^2(P_0) = kP_0$, then $\phi(P_0) = \pm wP_0$ for a square root $w$ of $k$ (or $q$) modulo $\ell$. If no such $w$ exists (if $k$ is a quadratic non-residue mod$\ell$) or if $\phi(P_0) \neq \pm wP_0$, then we must be in the subcase $\phi^2(P_0) = -kP_0$. Since testing the existence of a point $P_0 \in E[\ell] \setminus P_\infty$ such that $\phi(P_0) = \pm wP_0$ follows the same ideas as testing the existence of a point $P_0 \in E[\ell] \setminus P_\infty$ such that $\phi^2(P_0) = \pm kP_0$, we get the two following tests:

**Test 2.20** *Try to compute* $w \in \{1, 2, \ldots \frac{\ell-1}{2}\}$ *such that* $q \equiv w^2 \bmod \ell$. *If* $w$ *does not exist, then* $t_\ell = 0$.

**Test 2.21** *For* $w$ *satisfying Test 2.20, determine if*

$$\gcd\left((X^q - X)\Psi_w^2 + \Psi_{w-1}\Psi_{w+1}, f_\ell\right) \begin{cases} \neq 1 & (t_\ell = \pm 2w) \\ = 1 & (t_\ell = 0) \end{cases}.$$

If $\phi(P_0) = \pm wP_0$, we check if $\phi(P_0) = wP_0$ by matching the $y$ coordinates (if not, we have $\phi(P_0) = -wP_0$). This is equivalent to whether or not

$$Y_0^q = \frac{\Psi_{w+2}(P_0)\Psi_{w-1}^2(P_0) - \Psi_{w-2}(P_0)\Psi_{w+1}^2(P_0)}{4Y_0\Psi_w^3(P_0)} \tag{2.10}$$

for some point of order $\ell$. Once again, $\Psi_w(P) \neq 0$ for all $P \in E[\ell] \setminus P_\infty$, and we also have $\mathcal{Y}(P) \neq 0$ ($\mathcal{Y}(P) = 0 \iff 2P = P_\infty$ which is impossible if $P$ is a point of odd order $\ell$). We can therefore multiply the function of $X$ and $Y$ corresponding to (2.10) by $4Y\Psi_w(X, Y)$ and check whether or not we have an equality for some point of order $\ell$, that is if

$$4Y^{q+1}\Psi_w^3(X,Y) - \Psi_{w+2}(X,Y)\Psi_{w-1}^2(X,Y) + \Psi_{w-2}(X,Y)\Psi_{w+1}^2(X,Y)$$

has a common root with $\Psi_\ell(X, Y)$. This can be done via the following test:

**Test 2.22** *For* $w$ *satisfying Test 2.21, determine if*

$$\gcd\left(4Y^{q+1}\Psi_w^3 - \Psi_{w+2}\Psi_{w-1}^2 + \Psi_{w-2}\Psi_{w+1}^2, f_\ell\right) \begin{cases} \neq 1 & (t_\ell = 2w) \\ = 1 & (t_\ell = -2w) \end{cases}.$$

**Remark 2.23** The different tests are building up on each other. For example, a point of $E[\ell] \setminus P_\infty$ satisfying Test 2.22 must also satisfy Tests 2.19 and 2.21. To improve the efficiency of Tests 2.21 and 2.22, we can replace $f_\ell$ by the gcd found in Test 2.19. Once we make this restriction, Tests 2.21 and 2.22 can be read as saying "Is the polynomial congruent to zero modulo the factor of $f_\ell$ found in Test 2.19?"

**Remark 2.24** The subcase $\phi^2(P_0) = -kP_0$ is an "all-or-nothing" situation, i.e. either $\phi^2(P) = -kP$ for all $P \in E[\ell] \setminus P_\infty$ or $\phi^2(P) \neq -kP$ for all $P \in E[\ell] \setminus P_\infty$ (the existence of $P_0 \in E[\ell] \setminus P_\infty$ such that $\phi^2(P_0) = -kP_0$ forces $t_\ell = 0$, and then $\phi^2(P) + kP = P_\infty$ for all other $P \in E[\ell] \setminus P_\infty$). On the other hand, $\phi^2(P) = kP$ may be true for a nontrivial subset of $E[\ell]$, so if the gcd in Test 2.19 returns a nontrivial factor of $f_\ell$ then Test 2.21 can be skipped. Unfortunately, this improvement has an almost insignificant impact on the overall complexity of the algorithm.

**2.3.4 Case 2.** If $\phi^2(P) \neq \pm kP$ for every $P \in E[l] \setminus P_\infty$, then we can use the general addition formula of Section 2.1.2 to compute $\phi^2(P) + kP$ in terms of the $x$ and $y$ coordinates of $P$. To find $t_\ell$, we look for a match of the form $\tau\phi(P)$ with $\tau \in \{1, 2, 3, \ldots, \ell - 1\}$.

**Remark 2.25** Once again we have an "all-or-nothing" situation since $\phi^2(P) + kP = t_\ell\phi(P)$ for every $P \in E[\ell] \setminus P_\infty$. If we had $\phi^2(P_0) + kP_0 = \tau\phi(P_0)$ for some $P_0 \in E[\ell] \setminus P_\infty$ and $\tau \not\equiv t_\ell \bmod \ell$, then we would find $(t_\ell - \tau)\phi(P_0) = P_\infty$ which is impossible since $\phi(P_0)$ is also in $E[\ell] \setminus P_\infty$.

The $x$ and $y$ coordinates for $\phi^2(P) + kP$ are

$$\left(-X^{q^2} - X + \frac{\Psi_{k+1}\Psi_{k-1}}{\Psi_k^2} + \lambda_k^2, -Y^{q^2} + \lambda_k\left(2X^{q^2} + X - \frac{\Psi_{k+1}\Psi_{k-1}}{\Psi_k^2} - \lambda_k^2\right)\right)$$

with

$$\begin{aligned}\lambda_k &= \frac{\left(\frac{\Psi_{k+2}\Psi_{k-1}^2 - \Psi_{k-2}\Psi_{k+1}^2}{4Y\Psi_k^3}\right) - Y^{q^2}}{\left(X - \frac{\Psi_{k+1}\Psi_{k-1}}{\Psi_k^2}\right) - X^{q^2}} \\ &= \frac{\Psi_{k+2}\Psi_{k-1}^2 - \Psi_{k-2}\Psi_{k+1}^2 - 4Y^{q^2+1}\Psi_k^3}{4Y\Psi_k\left(\left(X - X^{q^2}\right)\Psi_k^2 - \Psi_{k+1}\Psi_{k-1}\right)}\end{aligned}$$

(recall that for all the points in $E[\ell] \setminus P_\infty$, $\Psi_k(P) \neq 0$ and $\mathcal{Y}(P) \neq 0$). To simplify the formulas, we will write $\lambda_k$ as $A_k/B_k$ with

$$\begin{aligned}A_k &= \Psi_{k+2}\Psi_{k-1}^2 - \Psi_{k-2}\Psi_{k+1}^2 - 4Y^{q^2+1}\Psi_k^3 \\ B_k &= 4Y\Psi_k\left(\left(X - X^{q^2}\right)\Psi_k^2 - \Psi_{k-1}\Psi_{k+1}\right)\end{aligned}$$

Note that $B_k(X, Y)$ is invertible modulo $\Psi_\ell(X, Y)$ since we found a gcd of 1 in Test 2.19.

For $\tau\phi(P) = \phi(\tau P)$, we get the coordinates

$$\left(\left(X - \frac{\Psi_{\tau+1}\Psi_{\tau-1}}{\Psi_\tau^2}\right)^q, \left(\frac{\Psi_{\tau+2}\Psi_{\tau-1}^2 - \Psi_{\tau-2}\Psi_{\tau+1}^2}{4Y\Psi_\tau^3}\right)^q\right)$$

which is equal (in $E[\ell]$) to

$$\left(X^q - \frac{\Psi_{\tau+1}^q\Psi_{\tau-1}^q}{\Psi_\tau^{2q}}, \frac{\Psi_{\tau+2}^q\Psi_{\tau-1}^{2q} - \Psi_{\tau-2}^q\Psi_{\tau+1}^{2q}}{4Y^q\Psi_\tau^{3q}}\right).$$

Also note that for $\tau \not\equiv 0 \bmod \ell$, $\Psi_\tau(P) \neq 0$ and $\mathcal{Y}(P) \neq 0$ for every $P \in E[\ell] \setminus P_\infty$, so $\Psi_\tau(X,Y)$ and $Y$ are invertible modulo $\Psi_\ell(X,Y)$.

To look for the value of $\tau$ such that $\phi^2(P)+kP = \tau\phi(P)$ for every $P \in E[\ell]\setminus P_\infty$, we will first try to match the $x$ coordinates, and then see if the $y$ coordinates also match. This approach as two advantages: since the $x$ coordinates of $\tau\phi(P)$ and $-\tau\phi(P)$ are identical, we will be able to test the $x$ coordinates for the values of $\tau$ two at a time. Furthermore, the test for the $x$ coordinates will be somewhat less expensive than the one for the $y$ coordinates, giving one more reason to do the tests in this order.

To test for the equality of the $x$ coordinates, we are looking for an equality of the form

$$-X^{q^2} - X + \frac{\Psi_{k+1}\Psi_{k-1}}{\Psi_k^2} + \frac{A_k^2}{B_k^2} \equiv X^q - \frac{\Psi_{\tau+1}^q \Psi_{\tau-1}^q}{\Psi_\tau^{2q}} \bmod \Psi_\ell \ ,$$

and since the denominators are all invertible modulo $\Psi_\ell$, we can write it as

$$\left(\Psi_{k-1}\Psi_{k+1}B_k^2 - \left(X^{q^2} + X^q + X\right) B_k^2\Psi_k^2 + \Psi_k^2 A_k^2\right)\Psi_\tau^{2q} + \Psi_{\tau-1}^q\Psi_{\tau+1}^q B_k^2\Psi_k^2 \equiv 0$$

modulo $\Psi_\ell$. At this point, we notice that parts of the equality depend on $\tau$ (which will vary), while others depend on $k$ (and $q$) and can be viewed as fixed. We write these fixed parts as:

$$\begin{aligned} D_k &= B_k^2\Psi_k^2 \\ C_k &= \Psi_{k-1}\Psi_{k+1}B_k^2 - \left(X^{q^2} + X^q + X\right) D_k + \Psi_k^2 A_k^2 \ , \end{aligned}$$

and the condition for the $x$ coordinates to match becomes

$$C_k\Psi_\tau^{2q} + \Psi_{\tau-1}^q\Psi_{\tau+1}^q D_k \equiv 0 \bmod \Psi_\ell \ ,$$

from which we obtain:

**Test 2.26** *For $\tau \in \{1, 2, \ldots, \frac{\ell-1}{2}\}$ and $k \equiv q \bmod \ell$, determine if*

$$C_k\Psi_\tau^{2q} + \Psi_{\tau-1}^q\Psi_{\tau+1}^q D_k \begin{cases} \equiv 0 \bmod f_\ell & (t_\ell = \pm\tau) \\ \not\equiv 0 \bmod f_\ell & (t_\ell \neq \pm\tau) \ . \end{cases}$$

For the $y$ coordinates, the equality is

$$-Y^{q^2} + \frac{A_k}{B_k}\left(2X^{q^2} + X - \frac{\Psi_{k+1}\Psi_{k-1}}{\Psi_k^2} - \frac{A_k^2}{B_k^2}\right) \equiv \frac{\Psi_{\tau+2}^q\Psi_{\tau-1}^{2q} - \Psi_{\tau-2}^q\Psi_{\tau+1}^{2q}}{4Y^q\Psi_\tau^{3q}}$$

modulo $\Psi_\ell$, which is equivalent to:

$$\begin{aligned} 4Y^q \left(A_k \left(\left(2X^{q^2} + X\right) B_k^2\Psi_k^2 - \Psi_{k-1}\Psi_{k+1}B_k^2 - \Psi_k^2A_k^2\right) - Y^{q^2}B_k^3\Psi_k^2\right)\Psi_\tau^{3q} \\ - \left(\Psi_{\tau+2}^q\Psi_{\tau-1}^{2q} - \Psi_{\tau-2}^q\Psi_{\tau+1}^{2q}\right) B_k^3\Psi_k^2 \equiv 0 \end{aligned}$$

modulo $\Psi_\ell$. Once again, we can consider the terms dependent on $k$ (and $q$) as fixed, writing them as

$$\begin{aligned} F_k &= -B_k D_k \\ E_k &= 4Y^q\left(A_k\left(\left(X^{q^2} - X^q\right) D_k - C_k\right) + Y^{q^2} F_k\right) \ . \end{aligned}$$

The condition to have matching $y$ coordinates becomes

$$E_k\Psi_\tau^{3q} + \left(\Psi_{\tau+2}^q\Psi_{\tau-1}^{2q} - \Psi_{\tau-2}^q\Psi_{\tau+1}^{2q}\right) F_k \equiv 0 \bmod \Psi_\ell \ ,$$

giving us the last test needed for Schoof's algorithm:

**Test 2.27** *For $\tau$ satisfying Test 2.26 and $k \equiv q \bmod \ell$, determine if*

$$E_k \Psi_\tau^{3q} + \left(\Psi_{\tau+2}^q \Psi_{\tau-1}^{2q} - \Psi_{\tau-2}^q \Psi_{\tau+1}^{2q}\right) F_k \begin{cases} \equiv 0 \bmod f_\ell & (t_\ell = \tau) \\ \not\equiv 0 \bmod f_\ell & (t_\ell = -\tau) \end{cases} .$$

**Remark 2.28** Since Test 2.27 is used only once, using $E_k$ and $F_k$ is more a question of esthetic and readability than a question of efficiency. For Test 2.26 however, computing $C_k$ and $D_k$ and reusing them for every value of $\tau$ tested helps reduce the computational cost of the algorithm.

**Remark 2.29** An alternative to the approach used in Tests 2.26 and 2.27 would be to compute the polynomials describing the $x$ and $y$ coordinates of the point $(X^{q^2}, Y^{q^2}) + k(X,Y)$ (modulo $\Psi_\ell(X,Y)$) and compare those with the polynomials for the points $\tau\phi(X,Y)$ obtained by repeatedly adding $(X^q, Y^q)$ to itself (again modulo $\Psi_\ell(X,Y)$) using the formulas of Section 2.1.2, i.e. comparing $\phi^2(P) + kP$ with $\tau\phi(P)$ for $P \in E[\ell] \setminus P_\infty$.

This second approach has the advantage of reducing the number of exponentiations modulo $\Psi_\ell(X,Y)$ to only four ($X^q$, $Y^q$, $X^{q^2}$ and $Y^{q^2}$) to complete all the tests required for case 2, but at the cost of computing a modular inverse for each value of $\tau$ tested and one for the computation of $(X^{q^2}, Y^{q^2}) + k(X,Y)$. We will see in Section 2.4.1 why Tests 2.26 and 2.27 are in fact more efficient.

**2.3.5 Algorithm.** The basic process of Schoof's algorithm can be seen in Algorithm 1. Note that in Case 2, we compute the polynomials $\Psi_i^q \bmod \Psi_\ell$ before doing any of the tests, the reason for this will be explained in Section 2.4.1. All other practical improvements presented in Section 2.4 are not included in Algorithm 1.

**2.3.6 Asymptotic Running Time.** We can now discuss the asymptotic running time of Schoof's algorithm. For this, we will assume that fast arithmetic algorithms are used, i.e.

- Elementary field arithmetic operations (addition, subtraction, multiplication and division) can be done in $O(\log(q)\log\log(q)\log\log\log(q)) = O(\log^{1+\epsilon}(q))$ bit operations.
- Elementary polynomial operations (addition, subtraction, multiplication, squaring, reduction, and gcd) can be done in $O(d\log(d)\log\log(d)) = O(d^{1+\epsilon})$ field operations for polynomials of degree $\leq d$.

These assumptions are too optimistic for practical cases (for example when $q$ is a 160-bits prime), which will be addressed in Section 2.4.

Since $L$ is the smallest prime such that $\prod_{\ell \text{ prime}, \ell \leq L} \ell \geq 4\sqrt{q}$, there will be $O(\log(q))$ primes and, using the prime number theorem, $L = O(\log^{1+\epsilon}(q))$ (the $+\epsilon$ could be removed from the exponent using other number theoretic results, but this is not essential here since arithmetic costs also contain a $+\epsilon$ in the exponent).

The largest polynomial handled will have degree $d = (L^2-1)/2$, so $d = O(\log^{2+\epsilon}(q))$ and all polynomial operations (other than exponentiation) can then be assumed to require $O(\log^{3+\epsilon}(q))$ bit operations. From this, computing the list of $\Psi_i$ for $i$ up to $L$ will require $O(\log^{4+\epsilon}(q))$ bit operations.

In general, taking a polynomial to the power $q$ modulo $\Psi_\ell$ (i.e. computing $X^q$, $Y^q$, $X^{q^2}$ and $Y^{q^2}$ modulo $\Psi_\ell$) takes $O(\log^{4+\epsilon}(q))$ bit operations. For a given prime

**Algorithm 1** Schoof's algorithm for the elliptic curve $E$ over $\mathbb{F}_q$

**Inputs:** A nonsingular elliptic curve $E$ given by an equation $Y^2 = X^3 + aX + b$ over the field $\mathbb{F}_q$ of large characteristic
**Outputs:** $|E(\mathbb{F}_q)|$
Let $L$ be the smallest prime such that $\prod_{\ell \text{ prime}, \ell \leq L} \ell \geq 4\sqrt{q}$
**for** $i = 0$ to $L$ **do**
  Compute the division polynomial $\Psi_i$
**end for**
Run Test 2.18 to get $t_2 \equiv t \bmod 2$
**for** $\ell$ prime from 3 to $L$ **do**
  Compute $X^q$, $X^{q^2}$, $Y^q$, and $Y^{q^2}$ modulo $\Psi_\ell$
  Run Test 2.19 to find if $\ell$ is in Case 1 or Case 2
  **if** $\ell$ is in Case 1 **then**
    Run Test 2.20 to find a square root $w$ of $q$ mod $\ell$ (if possible)
    **if** $w$ exists **then**
      Run Test 2.21 to see if $t_\ell = \pm 2w$ or $t_\ell = 0$
      **if** $t_\ell = \pm 2w$ **then**
        Run Test 2.22 to find the sign of $t_\ell$
      **end if**
    **end if**
  **else**
    Compute $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ and $F_k$ modulo $\Psi_\ell$
    **for** $i = 0$ to $\ell - 1$ **do**
      Compute $\Psi_i^q$ mod $\Psi_\ell$
    **end for**
    **for** $\tau = 1$ to $(\ell - 1)/2$ while $t_\ell$ has not been found **do**
      Run Test 2.26 to see if $t_\ell = \pm\tau$
      **if** $t_\ell = \pm\tau$ **then**
        Run Test 2.27 to find the sing of $t_\ell$
      **end if**
    **end for**
  **end if**
**end for**
Use the Chinese Remainder Theorem to compute $t \in [-2\sqrt{q}, 2\sqrt{q}] \cap \mathbb{Z}$ such that $t \equiv t_\ell \bmod \ell$ for every prime $\ell \leq L$
**Return** $|E(\mathbb{F}_q)| = q - t + 1$

$\ell$, determining if the prime is in Case 1 or Case 2, and computing $t_\ell$ if $\ell$ is in Case 1 can then be done in $O(\log^{3+\epsilon}(q))$ bit operations since each one of the tests requires a constant number of polynomial operations given $X^q$, $Y^q$, $X^{q^2}$, and $Y^{q^2}$ modulo $\Psi_\ell$.

For the primes $\ell$ in Case 2, we also compute a number of other polynomials modulo $\Psi_\ell$ before doing Tests 2.26 and 2.27. Computing $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ and $F_k$ modulo $\Psi_\ell$ takes a constant number of polynomial operations, so $O(\log^{3+\epsilon}(q))$ bit operations. As we will see in Section 2.4.1, the complete set of $\Psi_i^q$ mod $\Psi_\ell$ can be computed in $O(\ell) = O(\log^{1+\epsilon}(q))$ polynomial operations, for a total of $O(\log^{4+\epsilon}(q))$ bit operations. Once these polynomials have been computed, each run of Tests 2.26

or Test 2.27 requires a constant number of polynomial operations, and we need at most $(\ell - 1)/2 = O(\log^{1+\epsilon}(q))$ such runs, for a total of $O(\log^{4+\epsilon}(q))$ bit operations.

Computing $t_\ell$ can therefore be done in $O(\log^{4+\epsilon}(q))$ bit operations for every $\ell$, and since there are $O(\log(q))$ of them, we have a total of $O(\log^{5+\epsilon}(q))$ bit operations.

Once this is done, computing $t$ with the Chinese Remainder Theorem takes a time $O(\log^{3+\epsilon}(q))$, which can be considered insignificant. The total running time is therefore $O(\log^{5+\epsilon}(q))$.

## 2.4 Implementation

In this section, we will look at how Schoof's algorithm can be implemented to minimize the running time as much as possible without introducing any new theory. The main issues we deal with are the following:

- *Reducing the cost associated to the exponentiation of polynomials:*
  Read in its original form, Schoof's algorithm requires taking a large number of polynomials to the power $q$ modulo $\Psi_\ell$. If these exponentiations are computed using a standard square-and-multiply algorithm, they very quickly become the dominant cost of the whole algorithm, introducing an extra multiple of $O(\log(q))$ in the analysis of Section 2.3.6 (in fact, this extra multiple appears independently of the type of polynomial arithmetic used).

  As we already mentioned, we will show in Section 2.4.1 how the exponentiations occurring in Tests 2.26 and 2.27 can be computed much more quickly than would be the case in general. We will also discuss what approach can be used for general polynomials.
- *Working with polynomials in one variable instead of two;*
  We first observe that most equations in Section 2.3 involve polynomials in two variables $X$ and $Y$ in the ring $\mathbb{F}_q[X,Y]/(Y^2 - X^3 - aX - b)$ (which can therefore be reduced to polynomials of the form $p_0(X)+p_1(X)Y$. Although this is very useful to establish the theory, working with polynomials in two variables is rather cumbersome in practice.

  With a little more theoretical work, we can completely avoid using polynomials of two variables. Looking at the equations in more details, we can see that all the polynomials in Section 2.3, including all intermediate products, are either of the form $p_0(X)$ or of the form $p_1(X)Y$. This is where the polynomials $f_i(X)$ become very useful, as $\Psi_n(X,Y)$ can be written either as $f_n(X)$ ($n$ odd) or $Yf_n(X)$ ($n$ even).

  We will then rewrite all the equations in terms of polynomials of $X$, breaking things down into cases corresponding to the number of multiples of $Y$ involved. For readability, we will still keep even powers of $Y$ in the formulas, with the understanding that those are really polynomials in $X$ via the substitution $Y^2 = X^3 + aX + b$.
- *Reducing the degree of the polynomials whenever possible:*
  For this, we rely on the observation that for every $P \in E[\ell]$ and $i \in \{1, \ldots, \ell - 1\}$, we have $iP = -(\ell - i)P$. When we apply Corollary 2.13 and Proposition 2.16 to compute the $x$ and $y$ coordinates of $iP$ (respectively $-(\ell - i)P$), we deal with relations involving polynomials $f_j(X)$ of degree close to $i^2/2$ (respectively $(\ell - i)^2/2$).

  Since the $x$ coordinates of $iP$ and $(\ell - i)P$ are the same when $P \in E[\ell]$, the formulas of Corollary 2.13 with $m = i$ and $m = \ell - i$ must be equivalent

modulo $\Psi_\ell$. Similarly, the $y$ coordinates of $iP$ and $(\ell - i)P$ are the negative of each other when $P \in E[\ell]$, so the formulas of Proposition 2.16 with $n = i$ and $n = \ell - i$ only differ by a factor of $-1$ modulo $\Psi_\ell$.

As the cost of polynomial multiplication is heavily dependent on the degree of the inputs, choosing the formulas with polynomials of smaller degree will reduce the computational cost. For the computations, we will then replace $i$ by $\ell - i$ in the scalar multiplications when $i > (\ell - 1)/2$ (adjusting the signs accordingly when we are looking at the $y$ coordinates).

We also note that we will not make any assumption on the speed of the polynomial arithmetic. Even though the product of two polynomials of degree $d$ can be computed in *asymptotic time* $O(d \log(d) \log\log(d))$ (or $O(d^{1+\epsilon})$ to simplify), in practice this is not always the case. In general, the best we can say is that the cost of multiplication will be between $O(d^{1+\epsilon})$ and $O(d^2)$. The exact cost will depend on the technique(s) used at a given degree, which in turn depends on the implementation of the polynomial arithmetic.

At the sizes we are dealing with in our implementation of Schoof's algorithm, it would be more realistic (although a little pessimistic) to assume that multiplication costs grow quadratically with the degree of the polynomials. This observation will affect how we do some of the computations, in particular when it comes to computing powers of polynomials in the next subsection.

**2.4.1 Frobenius.** In this section, we will look at how to compute $q$-th powers of the polynomials $f_k^q \bmod f_\ell$. In general, the fastest method to compute $P_1(X)^n \bmod P_0(X)$ is through the square-and-multiply approach, which takes $\log_2(n)$ squares and $\omega(n)\log_2(n)$ multiplications module $P_0(X)$, where $\omega(n)$ is the density of the binary representation of $n$ (which is between 0 and 1). When the polynomials are defined in a field $\mathbb{F}_q$ of characteristic $p$, the situation changes significantly when $n$ is a power of $p$, and particularly so when $n = q$. We will present two alternative methods to compute the $q$-th powers: the first one is general, and could be used to compute $X^{q^2}$ and $Y^{q^2-1}$ or in the context of Section 2.5.3, while the second one is specific to the computation of $f_k^q$.

If $P_0(X)$ is of degree $d$ and $P_1(X)$ is reduced modulo $P_0(X)$, then we can write

$$P_1(X) \equiv \sum_{i=0}^{d-1} p_i X^i$$

and, using the properties of the Frobenius ($q$-th powers in $\mathbb{F}_q$), $P_1(X)^q \bmod P_0(X)$ becomes

$$\begin{aligned} P_1(X)^q &= \sum_{i=0}^{d-1} (p_i X^i)^q \\ &= \sum_{i=0}^{d-1} p_i^q X^{iq} \\ &\equiv \sum_{i=0}^{d-1} p_i((X^q)^i \bmod P_0(X)) \ . \end{aligned} \tag{2.11}$$

This produces a simple alternative to the square-and-multiply approach: starting from $X^q \bmod P_0(X)$, compute all the polynomials $(X^q)^i \bmod P_0(X)$ (recursively multiplying by $X^q \bmod P_0(X)$) and substitute them in Equation 2.11.

In general, it may be too expensive to compute all the powers of $X^q$ up to $X^{q(d-1)}$. Instead, we can compute $X^q, (X^q)^2, \ldots, (X^q)^r$ modulo $P_0(X)$ for a selected value of $r$. We then write $P_1(X)$ as

$$P_1(X) = \sum_{j=0}^{\lceil \frac{d-1}{r} \rceil} Q_j(X) X^{rj}$$

where the $Q_j(X)$ have degree at most $r-1$. Since we know $X^q, \ldots, (X^q)^{r-1}$, we could use the substitution approach to compute $Q_j(X)^q \bmod P_0(X)$, and use multiplications by $X^{qr} \bmod P_0(X)$ to get $P_1(X)^q \bmod P_0(X)$ with

$$P_1(X)^q \equiv \sum_{j=0}^{\lceil \frac{d-1}{r} \rceil} Q_j(X)^q (X^{qr})^j \bmod P_0(X) \ .$$

This gives us Algorithm 2.

**Remark 2.30** Algorithm 2 assumes that $X^q \bmod P_0(X)$ is already known. This is in fact a very mild assumption, since we need $X^q$ for other parts of our computations, in particular to obtain $X^{q^2} \bmod f_\ell$ in Test 2.19.

---

**Algorithm 2** Computation of $P_1(X)^q \bmod P_0(X)$ with $r$ precomputations

---

**Inputs:** Polynomials $P_0(X)$ and $P_1(X)$ with $\deg(P_1) < \deg(P_0)$, $X^q \bmod P_0(X)$ and parameter $r$
**Outputs:** $P_1(X)^q \bmod P_0(X)$
 **for** $i = 2$ to $r$ **do**
  **if** $i$ even **then**
   $(X^q)^i \bmod P_0(X) \leftarrow \left((X^q)^{i/2} \bmod P_0(X)\right)^2 \bmod P_0(X)$
  **else**
   $(X^q)^i \bmod P_0(X) \leftarrow \left((X^q)^{i-1} \bmod P_0(X)\right) (X^q \bmod P_0(X)) \bmod P_0(X)$
  **end if**
 **end for**
 Write $P_1(X)$ as $\sum_{j=0}^{\lceil \frac{d-1}{r} \rceil} Q_j(X) X^{rj}$ with $\deg(Q_j) < r$
 Compute $Q_{\lceil \frac{d-1}{r} \rceil}(X)^q \bmod P_0(X)$ by substitution
 $P(X) \leftarrow Q_{\lceil \frac{d-1}{r} \rceil}(X)^q \bmod P_0(X)$
 **for** $j = \lceil \frac{d-1}{r} \rceil - 1$ down to 0 **do**
  Compute $Q_j(X)^q \bmod P_0(X)$ by substitution
  $P(X) \leftarrow (P(X) \cdot (X^{qr} \bmod P_0(X)) \bmod P_0(X)) + (Q_j(X)^q \bmod P_0(X))$
 **end for**
 **Return** $P(X)$

---

Clearly, the computations of $(X^q)^i \bmod P_0(X)$ can be stored and re-used if more than one polynomial is taken to the power $q$ modulo $P_0(X)$. Algorithm 2 will then become more interesting as the number of exponentiation increases, and the choice of $r$ will reflect that. The precomputation phase (i.e. computing the $(X^q)^i \bmod P_0(X)$) takes $\frac{1}{2}r$ squares and $\frac{1}{2}r$ multiplications modulo $P_0(X)$, after which each exponentiation requires $d - \frac{d}{r}$ scalar multiplications ($p_i \cdot ((X^q)^i \bmod P_0(X))$) and $\frac{d}{r}$ multiplications modulo $P_0(X)$ (multiplications by $X^{qr} \bmod P_0(X)$).

To choose the best value of $r$ to minimize the cost, we will ignore the term in $\frac{d}{r}$ in the number of scalar multiplications. Although the cost of multiplying a polynomial by a constant is non-zero, in practice it is significantly smaller than the cost of multiplying two polynomials together (or squaring a polynomial), and the error introduced in our optimization will be smaller than the one caused by rounding up $r$ to the nearest integer (since $r$ must be an integer, but the *ideal* choice may not be). An easy computation shows that the optimal choice of $r$ is when we raise $s$ polynomials $P_1(X), P_2(X), \ldots, P_s(X)$ to the power $q$ modulo $P_0(X)$ is of the form $r = \sqrt{\frac{1+c}{2}}\sqrt{ds}$, where the constant $c$ is the ratio between polynomial squaring and multiplication costs (in general, a good approximation is $c \approx 0.8$). The average cost per polynomial will then be of $d - \sqrt{\frac{2}{1+c}}\sqrt{\frac{d}{s}}$ scalar multiplications, $\frac{1}{2}\sqrt{\frac{1+c}{2}}\sqrt{\frac{d}{s}}$ squares and $\frac{3}{2}\sqrt{\frac{1+c}{2}}\sqrt{\frac{d}{s}}$ multiplications modulo $P_0(X)$. If $\frac{d}{s}$ is smaller than $(\log(q))^2$, this approach will reduce the cost of exponentiation compared with the square-and-multiply method.

To compute $f_k^q \bmod f_\ell$, we can do even better. We first observe that the second line of Equation 2.11 can also be read as

$$P_1(X)^q \equiv P_1(X^q) \bmod P_0(X) \ , \tag{2.12}$$

which allows us to re-write Proposition 2.17 as

1. $f_0^q = 0$,
2. $f_1^q = 1$,
3. $f_2^q = 2$,
4. $f_3(X)^q \equiv 3(X^q)^4 + 6a(X^q)^2 + 12bX^q - a^2 \bmod f_\ell(X)$,
5. $f_4(X)^q \equiv 4(X^q)^6 + 20a(X^q)^4 + 80b(X^q)^3 - 20a^2(X^q)^2 - 16abX^q - 32b^2 - 4a^3 \bmod f_\ell(X)$.

The remaining $f_k^q \bmod f_\ell$ can then be computed using the recurrence relations of Corollary 2.15. Writing the relations in terms of the $f_k(X)$ instead of the $\Psi_k(X,Y)$, we have, first for $f_{2n}$:

$$f_{2n}^q = \left(\frac{1}{2}f_{n+2}f_n^3 - f_{n+1}^3 f_{n-1}\right)^q = \frac{1}{2}f_{n+2}^q(f_n^q)^3 - (f_{n+1}^q)^3 f_{n-1}^q \ ,$$

then for $f_{2n+1}$, $n$ odd:

$$f_{2n+1}^q = \left(f_{n+2}f_n^3 - Y^4 f_{n+1}^3 f_{n-1}\right)^q = f_{n+2}^q(f_n^q)^3 - Y^{4q}(f_{n+1}^q)^3 f_{n-1}^q \ ,$$

and $n$ even:

$$f_{2n+1}^q = \left(Y^4 f_{n+2}f_n^3 - f_{n+1}^3 f_{n-1}\right)^q = Y^{4q} f_{n+2}^q(f_n^q)^3 - (f_{n+1}^q)^3 f_{n-1}^q \ .$$

We can then compute the polynomials $f_k^q \bmod f_\ell$ iteratively as we first need them: this is not a problem since testing the different values of $\tau$ in consecutive order means that we need one new $f_k^q$ at every run of Test 2.26 (as well as for the final Test 2.27). Then $f_{2n}^q \bmod f_\ell$ can be computed with 2 squares and 3 multiplications modulo $f_\ell$ while $f_{2n+1}^q \bmod f_\ell$ can be computed with 2 squares and 5 multiplications modulo $f_\ell$.

In terms of asymptotic running times, we can therefore compute all the required $f_k^q \bmod f_\ell$ in $O(\ell)$ polynomial operations, which gives us the cost stated in Section 2.3.6.

**2.4.2 Tests 2.19 and 2.21.** For integers $n$ and $j$, we define a function $\mathbf{test2.19}_{n,j}$ which computes

$$\mathbf{test2.19}_{n,j} \equiv (X^n - X)\Psi_j^2 + \Psi_{j+1}\Psi_{j-1} \bmod f_\ell \ .$$

The function takes as inputs $X^n \bmod f_\ell$, $j$, $Y^2$, $f_\ell$ and the (precomputed) list of $f_j$'s.

Note that in Section 2.3, Test 2.19 computes

$$\gcd(\mathbf{test2.19}_{q^2,k}, f_\ell)$$

while Test 2.21 computes

$$\gcd(\mathbf{test2.19}_{q,w}, f_\ell) \ .$$

In Section 2.5.2, we will also need to compute $\mathbf{test2.19}_{q,\tau}$ (modulo a factor of $f_\ell$) for various values of $\tau$.

In practice, the case $j = 1$ is handled without a function call to $\mathbf{test2.19}_{n,1}$ since

$$\mathbf{test2.19}_{n,1} \equiv (X^n - X)\Psi_1^2 + \Psi_0\Psi_2 \equiv (X^n - X) \bmod f_\ell$$

can be computed directly from $X^n \bmod f_\ell$. For $j > 1$, $\mathbf{test2.19}_{n,j}$ is evaluated differently depending on the parity of $j$:

$$\mathbf{test2.19}_{n,j} \equiv \begin{cases} (X^n - X)Y^2 f_j^2 + f_{j-1}f_{j+1} \bmod f_\ell & \text{if } j \text{ is even} \\ (X^n - X)f_j^2 + Y^2 f_{j-1}f_{j+1} \bmod f_\ell & \text{if } j \text{ is odd} \ , \end{cases}$$

and both cases require 3 multiplications and 1 square modulo $f_\ell$.

**Remark 2.31** In Test 2.19 we may have $k > (\ell - 1)/2$ (all other calls of $\mathbf{test2.19}_{n,j}$ have $j < (\ell - 1)/2$). If this happens, we can replace $k$ by $k - \ell$ (since multiplication by $k$ and $k - \ell$ are equivalent in $E[\ell]$), and take advantage of the identity $f_{j-\ell}(X) = -f_{\ell-j}(X)$ to call $\mathbf{test2.19}_{q^2,\ell-k}$ instead of $\mathbf{test2.19}_{q^2,k}$ (note that all the negative signs cancel out). This reduces the degree of the polynomials in the computations (since $0 < \ell - k < (\ell - 1)/2$), therefore improving the speed of the test.

**2.4.3 Test 2.20.** In practice, we first use the Jacobi function to check if $w$ is a quadratic residue modulo $\ell$. If $k$ is a quadratic residue modulo $\ell$, then for $w = 1, 2, \ldots, \frac{\ell-1}{2}$, we compute $w^2 \bmod \ell$, until we find $w^2 \equiv k \bmod \ell$ (this $w$ becomes the square root of $q \bmod \ell$ returned by Test 2.20. We do not need to go further than $\frac{\ell-1}{2}$ since if $w$ is a square root of $k$ modulo $\ell$, then $\ell - w$ also is a square root of $k$.

Alternatively (if we want to avoid using the Jacobi function), we could try every $w$ in $\{1, 2, \ldots, \frac{\ell-1}{2}\}$ (stopping if we get $w^2 \equiv k \bmod \ell$) and return 0 if $w^2 \not\equiv k \bmod \ell$ for every $w$, understanding that 0 will mean that $k$ is not a quadratic residue modulo $\ell$.

Both approaches take $O(\ell)$ operations modulo $\ell$.

**2.4.4 Test 2.22.** For integers $n$ and $j$, we define a function $\mathbf{test2.22}_{n,j}$ which computes

$$\mathbf{test2.22}_{n,j} \equiv 4Y^{n+2}\Psi_j^3 - \Psi_{j+2}\Psi_{j-1}^2 + \Psi_{j-2}\Psi_{j+1}^2 \bmod f_\ell \ .$$

The function takes as inputs $Y^n \bmod f_\ell$, $j$, $Y^2$, $f_\ell$ and the (precomputed) list of $f_j$'s.

Note that in Section 2.3, Test 2.22 computes

$$\gcd(\mathbf{test2.22}_{q,w}, f_\ell)$$

In Section 2.5.2, we will also compute $\mathbf{test2.22}_{q,\tau}$ (modulo a factor of $f_\ell$) for one value of $\tau$.

In practice, the case $j = 1$ is handled without a function call to $\mathbf{test2.22}_{n,1}$ since

$$\mathbf{test2.22}_{n,1} \equiv 4Y^{n+2}\Psi_1^3 - \Psi_3\Psi_0^2 + \Psi_{-1}\Psi_2^2 \equiv 4Y^2(Y^n - 1) \bmod f_\ell$$

can be computed directly from $Y^n \bmod f_\ell$ (we can also ignore the factor $4Y^2$ since it is coprime to $f_\ell$). For $j > 1$, $\mathbf{test2.22}_{n,j}$ is evaluated differently depending on the parity of $j$:

$$\mathbf{test2.22}_{n,j} \equiv \begin{cases} 4Y^n f_j \left(Y^2 f_j\right)^2 - f_{j+2}f_{j-1}^2 + f_{j-2}f_{j+1}^2 \bmod f_\ell & \text{if } j \text{ is even} \\ Y^2\left(4Y^n f_j f_j^2 - f_{j+2}f_{j-1}^2 + f_{j-2}f_{j+1}^2\right) \bmod f_\ell & \text{if } j \text{ is odd} \end{cases},$$

and both cases require 5 multiplications and 3 square modulo $f_\ell$. To save an extra multiplication, we ignore the factor $Y^2$ when $j$ is odd ($Y^2$ is coprime to $f_\ell$, so the result of the tests will remain the same).

**2.4.5 Case 2.** From their definition, $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ and $F_k$ are defined in the ring $\mathbb{F}_q[X,Y]/(Y^2 - X^3 - aX - b)$, but all computation are done over polynomials in $X$ modulo $f_\ell(X)$, so we have to see how they are computed in practice. We replace $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ and $F_k$ by polynomials $A$, $B$, $C$, $D$, $E$ and $F$ in $\mathbb{F}_q[X]$ defined as follows (eliminating factors of $Y$ whenever it is convenient):

$$A_k = \begin{cases} AY & \text{if } k \text{ is even} \\ AY^2 & \text{if } k \text{ is odd} \end{cases} \qquad B_k = \begin{cases} BY^2 & \text{if } k \text{ is even} \\ BY & \text{if } k \text{ is odd} \end{cases}$$

$$C_k = \begin{cases} CY^4 & \text{if } k \text{ is even} \\ CY^2 & \text{if } k \text{ is odd} \end{cases} \qquad D_k = \begin{cases} DY^4 & \text{if } k \text{ is even} \\ DY^2 & \text{if } k \text{ is odd} \end{cases}$$

$$E_k = \begin{cases} EY^6 & \text{if } k \text{ is even} \\ EY^3 & \text{if } k \text{ is odd} \end{cases} \qquad F_k = \begin{cases} FY^6 & \text{if } k \text{ is even} \\ FY^3 & \text{if } k \text{ is odd} \end{cases}.$$

This choice of definition is done to keep the equations in terms of the $\Psi_j(X,Y)$ and those in terms of the $f_j(X)$ as close to each other as possible.

We can then apply the formulas from Section 2.3.4 to write the polynomials $A$ to $F$ in terms of the polynomials $f_{k-2}$, $f_{k-1}$, $f_k$, $f_{k+1}$, $f_{k+2}$, $Y^2$, $X^q \bmod f_\ell$, $X^{q^2} \bmod f_\ell$, and $Y^{q^2-1} \bmod f_\ell$ (note that $Y^{q^2-1}$ can indeed be written as a polynomial in $X$ since $q^2 - 1$ is even). When $k$ is even, we get:

$$\begin{aligned}
A &= f_{k+2}f_{k-1}^2 - f_{k-2}f_{k+1}^2 - 4Y^{q^2-1}(Y^2 f_k)^2 f_k \\
B &= 4((X - X^{q^2})Y^2 f_k^2 - f_{k-1}f_{k+1})f_k \\
D &= Y^2(Bf_k)^2 \\
C &= (f_{k-1}f_{k+1}B^2 + (f_k A)^2) - (X^{q^2} + X^q + X)D \\
F &= -BD \\
E &= 4Y^{q-1}(A((X^{q^2} - X^q)D - C) + Y^{q^2-1}Y^2 F)
\end{aligned}$$

and when $k$ is odd:

$$\begin{aligned}
A &= f_{k+2}f_{k-1}^2 - f_{k-2}f_{k+1}^2 - 4Y^{q^2-1}f_k^2 f_k \\
B &= 4((X - X^{q^2})f_k^2 - Y^2 f_{k-1}f_{k+1})f_k \\
D &= (Bf_k)^2 \\
C &= Y^2(f_{k-1}f_{k+1}B^2 + (f_k A)^2) - (X^{q^2} + X^q + X)D \\
F &= -BD \\
E &= 4Y^{q-1}Y^2(A((X^{q^2} - X^q)D - C) + Y^{q^2-1}F).
\end{aligned}$$

When $k > (\ell - 1)/2$, we replace $k$ by $k - \ell$ (since multiplication by $k$ and $k - \ell$ are equivalent in $E[\ell]$), and take advantage of the identity $f_{j-\ell}(X) = -f_{\ell-j}(X)$ to write the polynomials $A$ to $F$ in terms of polynomials of smaller degree (since $0 < \ell - k < (\ell - 1)/2$), which reduces the cost of the multiplications. When $\ell - k$ is even, we get:

$$\begin{aligned}
A &= -f_{\ell-k-2}f_{\ell-k+1}^2 + f_{\ell-k+2}f_{\ell-k-1}^2 + 4Y^{q^2-1}(Y^2 f_{\ell-k})^2 f_{\ell-k} \\
B &= -4((X - X^{q^2})Y^2 f_{\ell-k}^2 - f_{\ell-k+1}f_{\ell-k-1})f_{\ell-k} \\
D &= Y^2(Bf_{\ell-k})^2 \\
C &= (f_{\ell-k-1}f_{\ell-k+1}B^2 + (f_{\ell-k}A)^2) - (X^{q^2} + X^q + X)D \\
F &= -BD \\
E &= 4Y^{q-1}(A((X^{q^2} - X^q)D - C) + Y^{q^2-1}Y^2 F)
\end{aligned}$$

and when $\ell - k$ is odd:

$$\begin{aligned}
A &= -f_{\ell-k-2}f_{\ell-k+1}^2 + f_{\ell-k+2}f_{\ell-k-1}^2 + 4Y^{q^2-1}f_{\ell-k}^2 f_{\ell-k} \\
B &= -4((X - X^{q^2})f_{\ell-k}^2 - Y^2 f_{\ell-k+1}f_{\ell-k-1})f_{\ell-k} \\
D &= (Bf_{\ell-k})^2 \\
C &= Y^2(f_{\ell-k-1}f_{\ell-k+1}B^2 + (f_{\ell-k}A)^2) - (X^{q^2} + X^q + X)D \\
F &= -BD \\
E &= 4Y^{q-1}Y^2(A((X^{q^2} - X^q)D - C) + Y^{q^2-1}F).
\end{aligned}$$

Depending on the parity of $k$ (or $\ell - k$ when $k > (\ell - 1)/2$), the six polynomials can be computed in 20 or 21 multiplications and 7 squares modulo $f_\ell$.

**2.4.6 Test 2.26.** For integers $n$ and $j$, we define a function $\mathbf{test2.26}_{n,j}$ which computes

$$\mathbf{test2.26}_{n,j} \equiv C\Psi_j^{2n} + \Psi_{j-1}^n \Psi_{j+1}^n D \bmod f_\ell \ ,$$

where $C$ and $D$ are defined from $C_n$ and $D_n$ as in Section 2.4.5. The function takes as inputs $C$, $D$, $j$, $Y^{2n} \bmod f_\ell$, $f_\ell$ and the (precomputed) list of $f_j^n$'s.

In practice, the case $j = 1$ is handled without a function call to $\mathbf{test2.26}_{n,1}$ since

$$\mathbf{test2.26}_{n,1} \equiv C\Psi_1^{2n} + \Psi_0^n \Psi_2^n D \equiv C \bmod f_\ell$$

which requires no computation if $C$ is known. For $j > 1$, $\mathbf{test2.26}_{n,j}$ is evaluated differently depending on the parity of $j$:

$$\mathbf{test2.26}_{n,j} \equiv \begin{cases} CY^{2n}(f_j^n)^2 + f_{j-1}^n f_{j+1}^n D \bmod f_\ell & \text{if } j \text{ is even} \\ C(f_j^n)^2 + Y^{2n} f_{j-1}^n f_{j+1}^n D \bmod f_\ell & \text{if } j \text{ is odd} \end{cases},$$

and both cases require 4 multiplications and 1 square modulo $f_\ell$.

**2.4.7 Test 2.27.** For integers $n$ and $j$, we define a function $\mathbf{test2.27}_{n,j}$ which computes

$$\mathbf{test2.27}_{n,j} \equiv E\Psi_j^{3n} + \left(\Psi_{j+2}^n \Psi_{j-1}^{2n} - \Psi_{j-2}^n \Psi_{j+1}^{2n}\right) F \bmod f_\ell \ ,$$

where $E$ and $F$ are defined from $E_n$ and $F_n$ as in Section 2.4.5. The function takes as inputs $E$, $F$, $j$, $Y^{2n} \bmod f_\ell$, $f_\ell$ and the (precomputed) list of $f_j^n$'s.

In practice, the case $j = 1$ is handled without a function call to $\mathbf{test2.27}_{n,1}$ since

$$\mathbf{test2.27}_{n,1} \equiv E\Psi_1^{3n} + \left(\Psi_3^n \Psi_0^{2n} - \Psi_{-1}^n \Psi_2^{2n}\right) F \equiv E + 4Y^{2n} F \bmod f_\ell$$

can be computed directly from $E$, $F$, and $Y^{2n} \bmod f_\ell$. For $j > 1$, $\mathbf{test2.27}_{n,j}$ is evaluated differently depending on the parity of $j$:

$$\mathbf{test2.27}_{n,j} \equiv EY^{2n}(f_j^n)^2 f_j^n + \left(f_{j+2}^n (f_{j-1}^n)^2 - f_{j-2}^n (f_{j+1}^n)^2\right) F \bmod f_\ell$$

if $j$ is even, and

$$\mathbf{test2.27}_{n,j} \equiv E(f_j^n)^2 f_j^n + \left(f_{j+2}^n (f_{j-1}^n)^2 - f_{j-2}^n (f_{j+1}^n)^2\right) Y^{2n} F \bmod f_\ell$$

if $j$ is odd. Both cases require 6 multiplications and 3 square modulo $f_\ell$.

**2.4.8 Using higher degrees of smaller primes.** If $\ell$ is one of the smaller primes, it might become interesting to compute $t \bmod \ell^n$ for $n > 1$ instead of using larger primes. For example, if the smallest unused prime is 73 and if we are missing a factor between 5 and 7 in order to have a product of the primes that is $\geq 4\sqrt{q}$, then it would be faster to compute $t \bmod 7^2$ rather than $t \bmod 73$ since we would be using $f_{49}(X)$ which as degree 1200 instead of degree 2664 for $f_{73}(X)$. The knowledge of $t \bmod 7$ gives us some further advantages:

1. The number of values to try is reduced ($t_{49}$ must satisfy $t_{49} \equiv t_7 \bmod 7$).
2. We actually work modulo a polynomial of degree 1176 instead of 1200 for $f_{49}(X)$ (admittedly not such a great improvement when we compare with degree 2664 for $f_{73}(X)$, but still helpful).

In general, we proceed as follows for $\ell^n$, $n > 1$:

- we first compute $t_{\ell^{n-1}} \equiv t \bmod \ell^{n-1}$;
- if $\tau \equiv t_{\ell^n} \bmod \ell^n$, then $\tau \equiv t_{\ell^{n-1}} \bmod \ell^{n-1}$, so there are at most $\ell$ possible values for $\tau$;
- if $\ell > 2$, we never need to test the $y$ coordinate (if a value passes the tests for the $x$ coordinate, the sign of $\tau$ is obtained from the previous statement);
- we must work over $E[\ell^n] \setminus E[\ell^{n-1}]$ since the points in $E[\ell^{n-1}]$ have order $\ell^{n-1}$, so we use

$$\tilde{f}_{\ell^n}(X) = \frac{f_{\ell^n}(X)}{f_{\ell^{n-1}}(X)}$$

which has degree $\ell^{2n-2}(\ell^2 - 1)/2$

Most of the process is done exactly as before, except that we only test possible values of $\pm\tau$. In particular, for case 2 we only test those values of $\tau$ for which $\tau \equiv \pm t_{\ell^{n-1}} \bmod \ell^{n-1}$ ($\tau \leq \frac{\ell^n-1}{2}$ as before).

**Remark 2.32** Special care should be taken when $t \equiv q+1 \bmod \ell$, i.e. when $\ell$ divides the group order. In this case, we know that $\gcd(X^q - X, \Psi_\ell) \neq 1$ since at least some of the $\ell$-torsion points are in $E(\mathbb{F}_q)$. If $\gcd(X^q - X, \Psi_\ell) \neq \Psi_\ell$ we can proceed as described above (there will be exactly $\ell - 1$ points of order $\ell$ in $E(\mathbb{F}_q) \setminus P_\infty$), but the situation is different when $\gcd(X^q - X, \Psi_\ell) = \Psi_\ell$ (i.e. when $X^q - X \equiv 0 \bmod \Psi_\ell$).

In the later case, **all** the $\ell$-torsion points are in $E(\mathbb{F}_q)$, which gives us $t \equiv q+1 \bmod \ell^2$ without any more computations (that is, the group order is divisible by $\ell 2^2$). To use the $\ell^2$-torsion points to gain more information on $t$ would then require a careful treatment of a number of possible special cases. To avoid doing this, we will only use the information $t \equiv q+1 \bmod \ell^2$ and not look for the value of $t$ modulo $\ell^3$. This is a reasonable compromise for the implementation of the algorithm since these curves should not be very common (close to 1 in $\ell^2$ on average), and are usually less interesting from a cryptographic perspective.

## 2.5 Improvements by Atkin and Elkies

From this point on, the reader should refer to the double paper by Schoof [75] and Morain [68] for more details about the theory. The presentation will be mostly expository on how to do the various computations.

To improve the running time of Schoof's algorithm, we rely on the following proposition of Atkin (see [75] for the proof):

**Proposition 2.33** *Let $E$ be a non-singular elliptic curve over $\mathbb{F}_q$ with $j$-invariant $j \neq 0$ or 1728. Let $\Phi_\ell(j,T) = g_1(T)g_2(T) \cdot g_s(T)$ be the factorization of the modular equation for $\ell$ (or, in practice, an equivalent polynomial) as a product of irreducible polynomials. The degrees of $g_1(T), g_2(T), \ldots, g_s(T)$ characterize the prime $\ell$ and they will form one of the following three types of distributions:*

(i) $1, \ell$
*Then $t_\ell^2 \equiv 4q \bmod \ell$ and $\phi$ acts on $E[\ell]$ as scalar multiplication by $\sqrt{q} \bmod \ell$. This can be considered a special case of type (iii).*

(ii) $r, r, r, \ldots, r$ $\quad r > 1$
*Then $t_\ell^2 - 4q$ is not a square modulo $\ell$ and $\phi$ acts on $E[\ell]$ as a $2 \times 2$ matrix with a characteristic polynomial irreducible modulo $\ell$. These $\ell$'s are referred to as Atkin primes.*

(iii) $1, 1, r, r, r, \ldots, r$
*Then $t_\ell^2 - 4q$ is a square modulo $\ell$ and (for a well selected pair of generators of $E[\ell]$) $\phi$ acts on $E[\ell]$ as a matrix $\begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix}$ with $\lambda, \mu \in \mathbb{F}_\ell^*$. These $\ell$'s are referred to as Elkies primes.*

Given the modular equation $\Phi_\ell(S,T)$ and the $j$-invariant of the elliptic curve $E$, we can determine if $\ell$ is an Atkin or an Elkies prime by computing

$$\gcd(\Phi_\ell(j,T), T^q - T)$$

and looking at the degree of the gcd.

**2.5.1 Atkin primes.** Since $\phi$ cannot act linearly on any subgroup of $E[\ell]$ (otherwise the characteristic polynomial of $\phi$ would split over $\mathbb{F}_\ell$), then case 1 (Section 2.3.3) only occurs when $t_\ell \equiv 0 \bmod \ell$. We still have to do Test 2.19, but none of the tests in case 1 are necessary. Since the case $t_\ell = 0$ applies to all the $\ell$-torsion points at once, we do not need to compute the gcd in Test 2.19 and simply verify that the remainder is 0 modulo $f_\ell$.

If $t_\ell \not\equiv 0 \bmod l$, then we go through case 2, with the following variation: before doing Test 2.26 for $\tau$, we check that $\tau^2 - 4q$ is not a square modulo $\ell$ (on average, this should remove the need for half of the runs of Test 2.26).

**2.5.2 Elkies primes.** For Elkies primes, the characteristic polynomial of $\phi$ factors over $\mathbb{F}_\ell$, which means that the $2 \times 2$ matrix associated to $\phi$ has two eigenvalues ($\lambda$ and $\mu$) and two eigenspaces ($C_\lambda$ and $C_\mu$) defined over $\mathbb{F}_\ell$.

The method introduced by Elkies construct an isogenous curve and uses it to find one of the eigenspaces $\phi$ and $g(X)$, the factor of $f_\ell(X)$ which corresponds to the points in this eigenspace. Using $g(X)$, we can then find the eigenvalue $\lambda$ corresponding to the eigenspace, and since the characteristic polynomial of $\phi$ is of the form $\phi^2 - t_\ell\phi + q = (\phi - \lambda)(\phi - \mu)$, we have that $\mu \equiv q/\lambda \bmod \ell$ and $t_\ell \equiv \lambda + \mu \bmod \ell$.

To do this, we start from the following proposition (see [75] for the proof):

**Proposition 2.34** *Let $E$ be a non-supersingular elliptic curve over $\mathbb{F}_q$ with $j$-invariant $j \neq 0$ or 1728. Then*

(i) *The polynomial $\Phi_\ell(j,T)$ has a zero $\tilde{j} \in \mathbb{F}_{q^r}$ if and only if the kernel $C$ of the corresponding isogeny $E \to E/C$ is a 1-dimensional eigenspace of $\phi^r$ in $E[\ell]$.*

(ii) *The polynomial $\Phi_\ell(j,T)$ splits completely in $\mathbb{F}_{q^r}$ if and only if $\phi^r$ acts as scalar multiplication on $E[\ell]$.*

For Elkies primes, we use the first part of Proposition 2.34 for zeros in $\mathbb{F}_q$. Since $\Phi_\ell(j,T)$ has two roots, $\tilde{j}_\lambda$ and $\tilde{j}_\mu$, in $\mathbb{F}_q$, each one is associated to a 1-dimensional eigenspace of $\phi$, $C_\lambda$ and $C_\mu$.

Since both $C_\lambda$ and $C_\mu$ are valid choices for the eigenspace, we choose one of the two $\mathbb{F}_q$-rational roots of $\Phi_\ell(j,T)$, call it $\tilde{j}$, and proceed as follows:

1. Using a series of identities on modular forms, we compute the equation of the isogenous curve $\tilde{E}$ and the constant term of $g(X)$ (Section 2.7).
2. From the Weierstraß functions $\mathcal{P}(z)$ (for $E$) and $\tilde{\mathbb{P}}(z)$ of $\tilde{E}$, and that $g(X)$ is defined by the $x$ coordinates of the points in $C$, the kernel of the isogeny between $E$ and $\tilde{E}$, we compute $g(X)$ (Section 2.8).
3. Find $\tau \in \{1, 2, \ldots, \frac{\ell-1}{2}\}$ such that $\mathbf{test2.19}_{q,\tau}$ is 0 modulo $g(X)$ (i.e. replacing $f_\ell(X)$ by $g(X)$ in the computations of $\mathbf{test2.19}_{q,\tau}$).
4. Using $\mathbf{test2.22}_{q,\tau}$, determine if the eigenvalue is $\tau$ or $-\tau$ (in the latter, replace $\tau$ by $\ell - \tau$).
5. $t_\ell \equiv \tau + q/\tau \bmod \ell$.

**2.5.3 Reducing the possibilities for $t \bmod \ell$.** In this section we elaborate (and correct an error) on the following proposition, which is the second part of Proposition 6.2 in [75] (the first part is our Proposition 2.33):

**Proposition 2.35** *If the nonlinear factors of $\Phi_\ell(j,T)$ have degree $r$, then*

$$t_\ell^2 \equiv (\zeta + \zeta^{-1} + 2)q \bmod \ell$$

*where $\zeta$, a primitive $r^{th}$ root of unity in $\overline{\mathbb{F}}_\ell$.*

**Proof** Let $\lambda$ and $\mu$ be the two eigenvalues (in $\overline{\mathbb{F}}_\ell$) of $\phi$ on $E[\ell]$, then $t \equiv \lambda + \mu$ and $\lambda\mu \equiv q$. Also, $\phi^r$ acts as scalar multiplication on $E[\ell]$ and $r$ is the smallest power of $\phi$ for which it does (Proposition 2.34). Then $\lambda^r \equiv \mu^r \bmod \ell$, so $q^r \equiv \lambda^r\mu^r \equiv \lambda^{2r} \bmod \ell$ and $\lambda^2 \equiv \zeta q \bmod \ell$ where $\zeta$ is a primitive $r^{\text{th}}$ root of unity in $\overline{\mathbb{F}}_\ell$. Note that $\zeta$ must be a primitive root since $\phi^s$ act as a linear matrix on $E[\ell]$ for every $s$ such that $\zeta^s \equiv 1$.

We will set $F_r(X)$ to be the minimal polynomial for all the primitive $r^{\text{th}}$ roots of unity in $\overline{\mathbb{F}_l}$. Then, $F_r(X)$ is given by

$$F_r(X) = \frac{X^r - 1}{\text{lcm}\{X^n - 1, n|r\}}$$

and if $\zeta$ is a primitive $r^{\text{th}}$ root of unity for $r > 2$, so is $\zeta^{-1}$ and

$$(X - \zeta)(X - \zeta^{-1}) \equiv X^2 - (\zeta + \zeta^{-1})X + 1 \bmod \ell$$

must divide $F_r(X)$ (since $\zeta \not\equiv \zeta^{-1}$). For $r > 2$, we have

$$\begin{aligned} t_\ell^2 &\equiv (\lambda + \mu)^2 \\ &\equiv \left(\lambda + \frac{q}{\lambda}\right)^2 \\ &\equiv \lambda^2 + 2q + \frac{q^2}{\lambda^2} \\ &\equiv \zeta q + 2q + \frac{q}{\zeta} \\ &\equiv (\zeta + \zeta^{-1} + 2)q \\ &\equiv (c + 2)q \end{aligned}$$

where $c$ is such that $X^2 - cX + 1$ divides $F_r(X)$. If $r = 2$, the process is the same but we have $\zeta = \zeta^{-1} = -1$, hence $t_\ell^2 = t_\ell = 0$ (for $r > 2$, $t_\ell$ cannot be zero). □

Combining Propositions 2.33 and 2.35, we obtain Algorithm 3 which can be used to reduce the cost of Schoof's algorithm.

Determining $r$ can be done by finding the smallest value of $i > 0$ such that

$$T^{q^i} - T \equiv 0 \bmod \Phi_\ell(j, T)$$

which is achieved by computing $T^{q^i} \equiv \left(T^{q^{i-1}}\right)^q \bmod \Phi_\ell(j, T)$ until $T^{q^i} \equiv T \bmod \Phi_\ell(j, T)$, in which case $r = i$. Note that the search can be stopped at $i = \frac{\ell+1}{2}$ for Atkin primes (respectively $\frac{\ell+1}{2}$ for Elkies primes) since there are no factors of $\ell + 1$ (resp. $\ell - 1$) between $\frac{\ell+3}{2}$ and $\ell$ (resp. $\frac{\ell+1}{2}$ and $\ell - 2$) and $r$ divides $\ell + 1$ (resp. $\ell - 1$). If the powers of $q$ are computed using square-and-multiply exponentiation, finding $r$ will require up to

$$\frac{\ell \pm 1}{2} \log(q) \cdot (1 \text{ square and } \omega \text{ multiplication})$$

of polynomials modulo $\Phi_\ell(j, T)$. If a substitution approach is used, we need up to

$$\frac{3(\ell \pm 1)}{4} \text{ multiplications} + \frac{\ell \pm 1}{4} \text{ squares} + \frac{(\ell \pm 1)^2}{2} \text{ scalar multiplications}$$

of polynomials modulo $\Phi_\ell(j, T)$, which is substantially faster, but still requires $\sim \frac{3}{2}\ell^3$ field operations if classical polynomial arithmetic is used for the multiplications.

**Algorithm 3** Reducing the number of possible values for $t_\ell = t \bmod \ell$

**Inputs:** $j$, $\ell$ and $\Phi_\ell(S,T)$.
**Outputs:** possible values for $\pm t_\ell$.

- Compute $\Phi_\ell(j,T)$
- Compute $T^q \bmod \Phi_\ell(j,T)$
- From $\gcd(\Phi_\ell(j,T), T^q - T)$, determine if $\ell$ is of type (i), (ii), or (iii)
- **if** $\ell$ is of type (ii) or (iii) **then**
  - Compute $r$ such that $T^{q^r} \equiv T \bmod \Phi_\ell(j,T)$
  - **if** $r = 2$ **then**
    - Return $t_\ell = 0$
  - **else**
    - $\mathcal{L} \leftarrow \emptyset$
    - $F_r(X) \leftarrow \frac{X^r - 1}{\mathrm{lcm}\{X^n - 1, n \mid r\}}$
    - **for** $c = 1$ to $\ell - 1$ **do**
      - **if** $X^2 - cX + 1$ divides $F_r(X)$ **then**
        - $d \leftarrow (c+2)q \bmod \ell$
        - **if** $\ell$ is of type (ii) **then**
          - **if** $d - 4q$ is not a quadratic residue mod $\ell$ **then**
            - $\mathcal{L} \leftarrow \mathcal{L} \cup \{\pm\sqrt{d}\} \bmod \ell$
          - **end if**
        - **else**
          - **if** $d - 4q$ is a quadratic residue mod $\ell$ **then**
            - $\mathcal{L} \leftarrow \mathcal{L} \cup \{\pm\sqrt{d}\} \bmod \ell$
          - **end if**
        - **end if**
      - **end if**
    - **end for**
    - Return $t_\ell \in \mathcal{L}$
  - **end if**
- **else**
  - Return $t_\ell \in \{\pm 2\sqrt{q}\} \bmod \ell$
- **end if**

For Atkin primes, we only need to test $\pm\tau$ for the values in the list. In practice this should reduce significantly the number of test to perform. We have:

$$\begin{aligned} \#\tau &\leq \frac{\phi(r)}{2} && \text{since we look at } \zeta + \zeta^{-1} \text{ for primitive } r^{\text{th}} \text{ roots of unity} \\ &\leq \frac{\phi(\ell+1)}{2} && \text{since } r \text{ divides } \ell + 1 \\ &\leq \frac{\ell+1}{4} && \text{since } \ell + 1 \text{ is even.} \end{aligned}$$

This reduces the number of test by half in the worst case scenario and by a larger factor in most cases.

In practice we do not use Proposition 2.35 to reduce the number of possible values of $t_\ell$ for Elkies primes since the time required to compute $r$ is slightly greater then the time saved by diminishing the number of cases to try (on average).

From an asymptotic point of view, computing $r$ could become interesting for larger values of $\ell$. Using the approach of Section 2.5.2, we do not look for $t_\ell$ directly, but rather for the eigenvalues associated to a (constructed) eigenspace $C$, so each potential candidate $\tau$ for the eigenvalue would have to satisfy that $\tau + q/\tau \bmod \ell$ is in the list of possible values of $t_\ell$.

**2.5.4 Choosing the small primes.** To maximize the savings from Atkin's and Elkies' improvements, we need to be selective when it comes to choosing the small primes used. Since the computations of $t_\ell$ for Elkies primes is done modulo a polynomial of degree $(\ell - 1)/2$ (i.e. $\ell + 1$ times smaller than the degree of $f_\ell$), the computational cost for Elkies primes will be significantly smaller than for Atkin primes of the same size. It will then become interesting to skip some Atkin primes even if it means going to a larger Elkies prime, where the computational cost of $t_\ell$ could be less.

The main problem then becomes: how to compare computations that require different numbers of polynomial operations modulo polynomials of different degrees, especially when the number of attempts to find $t_\ell$ will also behave as a random variable?

The solution we present here is by no mean a final, optimal, answer, but it has the advantage of being relatively close to the best possible choice while remaining simple to implement. We will rely on the following ideas:

- *Distribution of $t_\ell$:* We assume that possible values of $t_\ell$ (from Proposition 2.35) are distributed uniformly and independently in $\{1, 2, \ldots, \ell - 1\}$. If there are $n$ possible values for $t_\ell$, then each non-zero residue modulo $\ell$ has probability $n/(\ell - 1)$ of being a possible value for $t_\ell$. It also mean that if the residue $j$ is a possible value for $t_\ell$, then all other residues have probability $(n - 1)/(\ell - 2)$ of being a possible value for $t_\ell$, no matter if that residue is $j \pm 1$, $j \pm 2$, etc. The same argument applies to the eigenvalues with Elkies primes.
- *Number of tries to find $t_\ell$:* All the possible values for $t_\ell$ which satisfy Proposition 2.35 have the same probability of being $t_\ell$, and on average we have to try half of them to find the correct one. The same argument applies to the eigenvalues with Elkies primes.
- *Cost of polynomial operations:* The polynomial $f_k$ and the expressions used in the tests behave like random polynomials, i.e. their coefficients look like randomly chosen coefficients, there is no special abundance of zeros. We can then run experiments on randomly selected polynomials to evaluate the cost of polynomial multiplication, squaring, gcds, scalar multiplications, etc, when working modulo polynomials of degree $(\ell^2 - 1)/2$ (Atkin primes), $(\ell - 1)/2$ (Elkies primes) or $\ell \pm 1$ (Section 2.5.3). This observation will allow us to compare the costs of the computations of $t_\ell$ at the different primes $\ell$.

  We also note that prime field arithmetic is usually very stable when the (randomly chosen) primes are in the same range, for example when they have the same number of bit, or (to a lesser extent) when the number of bits are close to one another. This means that even if the implementation of the polynomial arithmetic takes into account changes in the ratios of the field arithmetic, we can use the cost estimates for the polynomial arithmetic on various prime fields of the same bit-size (rather than have to re-evaluate these cost for each new prime field).

- *Cost of determining if $\ell$ is an Atkin or an Elkies prime:* The cost of determining if $\Phi_\ell(j,T)$ has $\mathbb{F}_q$-rational roots or not is essentially independent from the value of $j$. We can therefore measure this cost experimentally for each prime $\ell$ (running the computation for some random values of $j$) and use it in our decision strategy. The same argument applies to the computation of $g_\ell(X)$ for Elkies primes.
- *Distribution of the Atkin and Elkies primes:* We will assume that every prime $\ell$ has a 50% change of being an Atkin prime and 50% change of being an Elkies prime for the elliptic curve $E$. This assumption will allow us to evaluate the probability (obviously 50%) that the next prime we look at is an Elkies prime.

Once $\ell$ is known to be an Elkies prime, we have to compute $g_\ell(X)$, then $X^q$ mod $g_\ell$ and $Y^{q-1}$ mod $g_\ell$, after which we run $\mathbf{test2.19}_{q,\tau}$ an average of $\ell/4$ times (to find $\pm t_\ell$) and $\mathbf{test2.22}_{q,\tau}$ once (to determine the sign). As well as the cost of computing $g_\ell$ (which is estimated using experiments), we have to take into account the cost of $\left(\frac{3}{4}\ell + 2\omega(q)\log_2(q) + 4\right)$ multiplications and $\left(\frac{1}{4}\ell + 2\log_2(q) + 2\right)$ squares modulo a polynomial of degree $(\ell-1)/2$.

If $\ell$ is known to be an Atkin prime, we first have to compute $r$ from Proposition 2.33. This requires the computation of $T^{q^i}$ mod $\Phi_\ell(j,T)$ recursively for $i$ going at most to $(\ell+1)/2$ (if $r$ is greater than $(\ell+1)/2$ then it must be equal to $\ell+1$). For our cost estimate, we will assume the worst case scenario where $i$ needs to go up to $(\ell+1)/2$. Using Algorithm 2 with $(\ell+1)/2$ precomputations (the optimal choice in this case), we get a total cost of $\frac{1}{2}(\ell-1)^2$ scalar multiplication, $\frac{3}{4}(\ell-1)$ multiplications, and $\frac{1}{4}(\ell-1)$ squares modulo a polynomial of degree $\ell+1$.

When $r$ is known, we can use Proposition 2.35 to reduce the number of possible values of $t_\ell$ by a factor of at least four (more in general). Since we first look at the $x$ coordinates, we then expect to test $\ell/16$ values before finding the right one (up to sign). We then have to perform Test 2.19 once, Test 2.26 an average of $\frac{\ell}{16}$ times and Test 2.27 once. These tests require the computation of the polynomials $A(X)$, $B(X)$, $C(X)$, $D(X)$, $E(X)$ and $F(X)$, as well as $X^q$ and $Y^{q-1}$ (computed using the square-and-multiply method), $Y^{2q}$ (computed as $(Y^{q-1})^2Y^2$) and $X^{q^2}$ and $Y^{q^2-1} = (Y^{q-1})^qY^{q-1}$ (which are computed with the substitution method).

Finally, we have to compute $f_j^q$ mod $f_\ell$ for a number of values of $j$. To compute these, we use the second method presented in Section 2.4.1, using the recurrence relations (which can be shown to be the fastest approach in this situation, although we omit the details here).

We evaluate the number of $j$'s as follows: On average, we will perform Test 2.26 up to $k = \ell/4$ and only for a quarter of the values between 1 and $k$. For the upper half of the interval, between $\ell/8$ and $\ell/4$, we need to do the test for $1/4$ of the values $j$, each of which requires $f_{j-1}^q$, $f_j^q$ and $f_{j+1}^q$. Taking into account the distribution, this means that each value of $j$ in that interval has a $\frac{37}{64}$ probability of being required. For the lower part of the interval, between 1 and $\ell/8$, we assume that we need $f_j^q$ for every $j$: not only they might be needed directly for the tests, but nearly all of them are also required to compute the $f_j^q$ in the upper half of the interval using the recurrence relations. Finally, we include the computation of $f_{k-2}^q$ and $f_{k+2}^q$ for Test 2.27, where $f_{j-2}^q$ has a $\frac{27}{64}$ probability of having already been computed. This gives us an estimate of $\frac{37}{128}\ell + \frac{101}{64}$ values of $j$.

We then have the following operations modulo a polynomial of degree $(\ell^2-1)/2$: $(\ell^2-\ell)$ scalar multiplications, $\left(\frac{315}{128}\ell + 2\omega(q)\log_2(q) + \frac{581}{16}\right)$ polynomial multiplications, $\left(\frac{267}{256}\ell + 2\log_2(q) + \frac{325}{32}\right)$ squares, all of which are done modulo a polynomial of degree $(\ell^2-1)/2$. We also have to compute the gcd of two polynomials of degree at most $(\ell^2-1)/2$. Combining this with the cost of finding $r$ gives us the total cost of computing $t_\ell$ for an Atkin prime.

For the higher powers of small primes, say $\ell^n$, the approach is essentially the same as for the Atkin primes, but the only tool available to reduce the number of values to test is the knowledge of $t \bmod \ell^{n-1}$. Since the $x$ coordinates are sufficient to determine $t \bmod \ell^n$ in this case (as there can be no doubt on the sign of $t$), we only have to perform Test 2.26 an average of $\ell/2$ times (saving the need for Tests 2.19 and 2.27, which also removes the need to compute $E(X)$ and $F(X)$). For the computation of the $f_j^q$, it is again possible to estimate the number of values required. Since the values for which we run Test 2.26 are placed at regular intervals, we need on average $\frac{1}{8}\ell^n + \frac{3}{4}\ell$ different $f_j^q$ including those needed for the recurrence relations (this assumes no overlap between the sets of $f_{j-1}^q$, $f_j^q$, $f_{j+1}^q$ in the upper half of the interval). We then have a total of $(\ell^{2n}-\ell^n)$ scalar multiplications, $(2\ell^n + 5\ell + 2\omega(q)\log_2(q) + 16)$ polynomial multiplications and $\left(\frac{3}{4}\ell^n + 2\ell + 2\log_2(q) + 7\right)$ squares, all of which are done modulo a polynomial of degree $\ell^{2n-2}(\ell^2-1)/2$ (i.e. modulo $f_{\ell^n}/f_{\ell^{n-1}}$).

We can then proceed as follows:

- The primes 2 and 3 are always used without looking at $\Phi_\ell$.
- The first few primes are used until we encounter the first Atkin prime. For all other primes $\ell$, we use them in increasing order of computational cost.
- We build a list of known Atkin primes that have not yet been used (in increasing order of cost).
- We determine if a prime is an Atkin or Elkies prime *only if* we are certain that it will be used if it is an Elkies prime (otherwise we use the cheapest Atkin prime, which is then removed from the list).
- Once a prime $\ell$ is determined to be an Atkin prime we estimate the cost of computing $t_\ell$ and add it to our list (if $\ell$ is an Elkies prime it is used right away and never goes into the list).
- Higher powers of small primes are treated as Atkin primes, they are automatically added to the list (without using $\Phi_\ell$ since that part of the theory does not apply). Whenever $\ell^n$ is used (including when $n=1$), we evaluate the cost of computing $t \bmod \ell^{n+1}$ and insert it in the list.
- To see if we want to determine whether or not $\ell$ is an Elkies prime, we use the following argument. Let $\tilde{\ell}$ be the cheapest prime in the list and $A_{\tilde{\ell}}$ be its estimated computational cost. Let $D_\ell$ be the cost of determining if $\ell$ is an Elkies prime or not, and $E_\ell$ be the estimated cost of computing $t_\ell$ if $\ell$ is an Elkies prime. We now have two choices: 1) We could use $\tilde{\ell}$ at a cost of $A_{\tilde{\ell}}$. 2) We could see if $\ell$ is an Elkies prime, be lucky 50% of the time and use $\ell$, or be unlucky the other 50% of the time and fall back to using $\tilde{\ell}$, for an "average" cost of $D_\ell + \frac{1}{2}(E_\ell + A_{\tilde{\ell}})$. We only consider using $\ell$ if $D_\ell + \frac{1}{2}(E_\ell + A_{\tilde{\ell}}) < A_{\tilde{\ell}}$, i.e. if $A_{\tilde{\ell}} > E_\ell + 2D_\ell$.
- We stop once the product of the small primes used is greater than $4\sqrt{q}$.

We can also introduce an upper bound to the costs in the list: if Schoof's algorithm (using all primes and, when useful, higher powers of small primes) would

terminate before a given $\ell$ (or $\ell^n$) is needed, then that $\ell$ (or $\ell^n$) will definitely not be needed if we choose to use some larger Elkies primes in our computations.

We could of course extend the argument $A_{\tilde{\ell}} > E_\ell + 2D_\ell$ to take into account that we may prefer to use the next prime (after $\ell$) if it is an Elkies prime rather than use $\tilde{\ell}$, or the next one after that, etc. Unfortunately, this quickly turns into a series of conditional loops that have to be solved on a case-by-case basis, which is why we decided to use the simpler approach (in any case, the difference in cost over the whole algorithm would be small).

## 2.6 Computing the modular equations

In Section 2.5, we ignored a very important problem: how to obtain the modular equation $\Phi_\ell$ for $\ell > 3$ (the machinery of Section 2.5 does not yield any computational savings for $\ell$ equal to 2 or 3). To do this, we rely heavily on the theory of modular forms, although we will omit most of the details. For the purpose of this text, we will only state the relevant results and show how they are used in the computations.

From a practical perspective, three options are available: we can either compute $\Phi_\ell \bmod p$ when we need it (without any long-term storage), precompute $\Phi_\ell \bmod p$ for a fixed $p$ and a range of $\ell$, or precompute $\Phi_\ell$ (over $\mathbb{Z}$) for a range of $\ell$ (reducing it modulo $p$ when it is used). Which one of these approaches is preferable depends on the situation where the group orders are computed.

In choosing which approach to use for the computation of $\Phi_\ell$, we must keep in mind a number of practical aspects, including the following:

- Precomputation costs can eventually be ignored if they are used for a large enough number of runs of the algorithm. If the precomputations are used a limited number of times, their cost should be taken into account for the running time of the implementation.
- The size of the integer coefficients of $\Phi_\ell$ grow very quickly as $\ell$ increases, so storing the modular equations $\Phi_\ell$ over $\mathbb{Z}$ is very expensive in terms of memory allocation.
- Due to the size of the coefficients, reducing the coefficients of $\Phi_\ell$ modulo $p$ cannot be considered completely free (even though it is less expensive than computing $\Phi_\ell \bmod p$ directly).
- Storing all the modular equations $\Phi_\ell \bmod p$ that we may use in the SEA algorithm takes $O(\log^{3+\epsilon}(q))$ elements in $\mathbb{F}_q$ ($O(\ell^2)$ for each $\Phi_\ell$). In comparison, computing $t_\ell$ for an Elkies prime $\ell$ can usually be done with $O(\log^{2+\epsilon}(q))$ elements of $\mathbb{F}_q$ in active memory (as long as there are enough small Elkies primes for the curve $E$).

As a general rule of thumb, we could consider the following approach:

- Fixed field: precomputations of $\Phi_\ell \bmod p$;
- Varying field, limited memory: computation of $\Phi_\ell \bmod p$ as it is needed;
- Varying field, large memory: precomputation of $\Phi_\ell$ over $\mathbb{Z}$.

Since our implementation was designed to compute group orders of elliptic curves for which the field size would be flexible (possibly even changing after every computation) on desktop computers (with access to sufficient memory), we decided to precompute the modular equations over $\mathbb{Z}$. As we are using a precomputation approach, we will not worry too much about the computational costs and look at techniques that are easier to implement rather than those optimized for the running

time. It should also be noted that computing $\Phi_\ell \bmod p$ directly (rather than first over $\mathbb{Z}$ and then reducing modulo $p$) is done essentially in the same way we will describe, but working over the ring (field) $\mathbb{Z}/p\mathbb{Z}$ instead of the ring $\mathbb{Z}$.

**2.6.1 Background.** Modular forms are functions of a variable $\tau$ is in the complex upper half-plane. For simplicity, they are usually written in terms of $\mathbf{q} = \exp(2\pi i\tau)$. We begin with the series $E_4$ and $E_6$ given in terms of $\mathbf{q}$ by

$$\begin{aligned} E_4(\mathbf{q}) &= 1 + 240 \sum_{n=1}^{\infty} \left( \sum_{k=1}^{\infty} n^3 \mathbf{q}^{nk} \right) \\ &= 1 + 240 \sum_{n=1}^{\infty} \frac{n^3 \mathbf{q}^n}{1 - \mathbf{q}^n} \end{aligned}$$

and

$$\begin{aligned} E_6(\mathbf{q}) &= 1 - 504 \sum_{n=1}^{\infty} \left( \sum_{k=1}^{\infty} n^5 \mathbf{q}^{nk} \right) \\ &= 1 + 504 \sum_{n=1}^{\infty} \frac{n^5 \mathbf{q}^n}{1 - \mathbf{q}^n} \ . \end{aligned}$$

Given $E_4(\mathbf{q})$ and $E_6(\mathbf{q})$, we associate to $\mathbf{q}$ the elliptic curve $E$ defined by the equation $Y^2 = X^3 + AX + B$ where $A = -E_4(\mathbf{q})/48$ and $B = E_6(\mathbf{q})/864$.

In the next sections, we will need the following modular functions:

$$\begin{aligned} \eta(\mathbf{q}) &= \mathbf{q}^{1/24} \prod_{n=1}^{\infty} (1 - \mathbf{q}^n) \\ \Delta(\mathbf{q}) &= \eta(\mathbf{q})^{24} = \mathbf{q} \prod_{n=1}^{\infty} (1 - \mathbf{q}^n)^{24} \\ j(\mathbf{q}) &= \frac{E_4(\mathbf{q})^3}{\Delta(\mathbf{q})}. \end{aligned}$$

The series expansion of $j(\mathbf{q})$ (the $j$-invariant of the elliptic curve defined by $\mathbb{Q}$) will have a great impact, so let us consider its coefficients. We observe that $E_4(\mathbf{q})$ can be written as

$$E_4(\mathbb{Q}) = 1 + \sum_{n=1}^{\infty} c_n \mathbf{q}^n$$

where $c_n$ is easily shown to lie between $240n^3$ and $300n^3$, while $1/\eta(\mathbb{Q})$ can be written as

$$\frac{1}{\eta(\mathbf{q})} = \frac{1}{\mathbf{q}^{1/24}} \sum_{n=0}^{\infty} p(n) \mathbf{q}^n$$

where $p(n)$ is the partition function. Asymptotically, $p(n)$ grows exponentially fast, and will be relatively close to $\exp(\pi\sqrt{2n/3})/(4n\sqrt{3})$. The coefficients of $j(\mathbf{q})$ will therefore grow as $O(n^3)$ when $n$ is small (say $n < 50$), but the exponential growth will quickly dominate.

To compute $\Phi_\ell$, we could use some of the properties of the modular equation:

- $\Phi_\ell(J,T)$ is of the form

$$\Phi_\ell(J,T) = J^{\ell+1} + T^{\ell+1} + \sum_{a=0}^{\ell}\sum_{b=0}^{\ell} s_{a,b} J^a T^b$$

  with $s_{\ell,\ell} = -1$.
- $\Phi_\ell$ is symmetric
- For a given $\mathbf{q}$, the roots of the polynomial (in $T$) $\Phi_\ell(j(\mathbf{q}),T)$ are $j(\mathbb{Q}^\ell)$ and $j(\mathbf{q}^{1/\ell}\zeta_\ell^k)$ for $k \in \{0,1,\ldots \ell-1\}$ (where $\zeta_\ell$ is a primitive $\ell$-th root of unity).

Note that the last property comes directly form Proposition 2.34.

Observing that $\Phi_\ell(j(\mathbf{q}), j(\mathbf{q}^\ell)) = 0$ for every $\mathbf{q}$ and that $j(\mathbf{q})$ has a pole of order 1 at $\mathbf{q} = 0$ while $j(\mathbf{q}^\ell)$ has a pole of order $\ell$, we could compute the coefficients $s_{a,b}$ as follows:

1. Compute the series $j(\mathbf{q})^{\ell+1}$, $j(\mathbf{q}^\ell)^{\ell+1}$ and $j(\mathbf{q})^a j(\mathbf{q}^\ell)^b$ for $0 \leq a,b \leq \ell$ at least up to their constant term (i.e. starting from the highest power of $1/\mathbf{q}$ to $\mathbf{q}^0$).
2. Set $S(\mathbf{q})$ as $S(\mathbf{q}) \leftarrow j(\mathbf{q}^\ell)^{\ell+1} + j(\mathbf{q})^{\ell+1} - j(\mathbf{q})^\ell j(\mathbb{Q}^\ell)^\ell$.
3. Working with decreasing values of $r = a + b\ell$ with $a \leq b \leq \ell$, $r < \ell^2$, we choose $s_{a,b}$ such that $S(\mathbf{q}) \leftarrow S(\mathbf{q}) + s_{a,b} j(\mathbf{q})^a j(\mathbf{q}^\ell)^b$ removes the pole of order $r$ at $\mathbf{q} = 0$ in $S(\mathbf{q})$ (if the pole at $\mathbf{q} = 0$ is of order less than $r$, we let $s_{a,b} = 0$). If $a < b$ we also let $s_{b,a} \leftarrow s_{a,b}$ and $S(\mathbf{q}) \leftarrow S(\mathbf{q}) + s_{b,a} j(\mathbf{q})^b j(\mathbf{q}^\ell)^a$ (to keep $\Phi_\ell$ symmetric).

The final value of $S(\mathbf{q})$ will then be $\Phi_\ell(j(\mathbf{q}), j(\mathbf{q}^\ell))$, and therefore must be the power series 0.

Unfortunately, the coefficients of $\Phi_\ell(J,T)$ grow extremely fast as $\ell$ increases, making it impractical to compute and store the polynomial for all the values of $\ell$ that we would need. This is mostly due to the degree of $j(\mathbf{q})$ in $\Phi_\ell(j(\mathbf{q}), j(\mathbf{q}^\ell))$: the power series of $j(\mathbf{q}^\ell)$ is obtained by replacing $\mathbf{q}^n$ by $\mathbf{q}^{n\ell}$ in the power series of $j(\mathbf{q})$ (whose coefficients grow very quickly). Once we have defined $S(\mathbf{q}) \leftarrow j(\mathbf{q}^\ell)^{\ell+1} + j(\mathbf{q})^{\ell+1} - j(\mathbf{q})^\ell j(\mathbb{Q}^\ell)^\ell$, $j(\mathbf{q})$ and $j(\mathbf{q}^\ell)$ appear only up to the power $\ell$, so the impact of the coefficients of powers of $j(\mathbf{q})$ will be greater than those of powers of $j(\mathbf{q}^\ell)$.

The exponential growth of the coefficients of $j(\mathbf{q})$ will quickly become a problem if we want to compute $\Phi_\ell$ over $\mathbb{Z}$. The remainder of this section will describe how to compute "equivalent" polynomials $P(J,T)$ whose degree in $J$ (i.e. $j(\mathbf{q})$) is much smaller, so the coefficients do not increase as quickly. Note that if we wanted to compute $\Phi_\ell \bmod p$ directly, the size of the coefficients would not be a problem as all the computations would be bounded by $p$, but the reduction of the degree in $J$ would still have a beneficial impact on the computational cost.

In practice, we try to restrict ourselves to do the computations using polynomials, or using truncated power series with integer powers of $\mathbf{q}$ and nonzero constant term (to make quotients of power series easier to handle). To do this, we will replace $\eta(\mathbf{q})$ with $\hat{\eta}(\mathbf{q})$ defined as:

$$\hat{\eta}(\mathbf{q}) = \prod_{n=1}^{\infty}(1-\mathbf{q}^n) = \eta(\mathbf{q})/\mathbf{q}^{1/24} \ .$$

Similarly, $\Delta(\mathbf{q})$ is replaced by $\hat{\Delta}(\mathbf{q})$ and $j(\mathbf{q})$ by $\hat{j}(\mathbf{q})$:

$$\hat{\Delta}(\mathbf{q}) = \prod_{n=1}^{\infty} (1-\mathbf{q}^n)^{24} = \Delta(\mathbf{q})/\mathbf{q} = \hat{\eta}(\mathbf{q})^{24}$$

$$\hat{j}(\mathbf{q}) = \frac{E_4(z)^3}{\hat{\Delta}(\mathbf{q})} = \mathbf{q}j(\mathbf{q}) \ .$$

Since these power series are used for all the values of $\ell$, their expansion is computed to the highest degree that will be required for the set of values of $\ell$ and are later truncated as desired when they are used for a specific $\ell$.

**2.6.2 Canonical case.** We first introduce a new modular function,

$$f(\mathbf{q}) = \ell^u \left(\frac{\eta(\mathbf{q}^\ell)}{\eta(\mathbf{q})}\right)^{2u} = \sum_{n=v}^{\infty} a_n \mathbf{q}^n \ ,$$

with

$$u = \frac{12}{\gcd(12, \ell-1)} \quad \text{and} \quad v = \frac{\ell-1}{\gcd(12,\ell-1)} \ .$$

Note that $u$ is defined in this way since it is the smallest positive integer $k$ such that the expansion of $\left(\ell\left(\frac{\eta(\mathbf{q}^\ell)}{\eta(\mathbf{q})}\right)^2\right)^k$ has only integer powers of $\mathbf{q}$, and such that the order of the pole at $\mathbb{Q}=0$ in $1/f(\mathbf{q})$ is kept as small as possible (the reason for this will become clear in Proposition 2.36).

Given a modular function $h(\mathbf{q})$, we will denote its image under the Atkin-Lehner involution by $h^*(\mathbf{q})$. For $j(\mathbf{q})$, we have $j^*(\mathbf{q}) = j(\mathbf{q}^\ell)$, and for $f(\mathbf{q})$ we have

$$f^*(\mathbf{q}) = \frac{\ell^u}{f(\mathbb{Q})} \ .$$

Just as $\Phi_\ell$ is defined such that the roots of $\Phi_\ell(j(\mathbf{q}),T)$ are $j(\mathbf{q}^\ell) = j^*(\mathbf{q})$ and $j(\mathbf{q}^{1/\ell}\zeta_\ell^k)$, we will define the polynomial $\Phi(J,T)$ such that the roots of $\Phi(j(\mathbf{q}),T)$ are at $f^*(\mathbf{q})$ and $f(\mathbf{q}^{1/\ell}\zeta_\ell^k)$. We observe that both $f(\mathbf{q})$ and $f^*(\mathbf{q})$ are power series with coefficients in $\mathbb{Z}$ from their definitions in terms of $\eta(\mathbf{q})$ and $\eta(\mathbf{q}^\ell)$. The coefficients of $f^*(\mathbf{q})$ are much smaller than those of $f(\mathbf{q})$, so we will in fact compute $f^*(\mathbf{q})$ first and compute $f(\mathbf{q})$ as $f(\mathbf{q}) = \ell^u/f^*(\mathbf{q})$.

This function will be used through the following proposition (see [68] for details):

**Proposition 2.36** *The polynomial $\Phi(J,T)$ defined by*

$$\Phi(j(\mathbf{q}),T) = (T - f^*(\mathbf{q})) \prod_{k=0}^{\ell-1} \left(T - f(\mathbf{q}^{1/\ell}\zeta_\ell^k)\right)$$

*(where $\zeta_\ell$ is a primitive $\ell$-th root of unity) can be used to replace $\Phi_\ell(J,T)$, i.e. for any given value $j$, the irreducible factors of the polynomials $\Phi_\ell(j,T)$ and $\Phi(j,T)$ have the same degrees.*

To compute $\Phi(J,T)$, we will first consider $\Phi(j(\mathbf{q}),T)$ as a polynomial in $T$ whose coefficients are power series in $\mathbf{q}$, and then turn those power series into polynomials

in $\mathbb{Z}[j(\mathbf{q})]$. To make the computations of the power series more manageable, we will first deal with the

$$\prod_{k=0}^{\ell-1}\left(T - f(\mathbf{q}^{1/\ell}\zeta_\ell^k)\right)$$

part, and then include the $(T - f^*(\mathbf{q}))$ factor. We write

$$\prod_{k=0}^{\ell-1}\left(T - f(\mathbf{q}^{1/\ell}\zeta_\ell^k)\right) = \sum_{r=0}^{\ell} c_r(\mathbf{q})T^{\ell-r}$$

where $c_0(\mathbb{Q}) = 1$, and

$$\Phi(j(\mathbf{q}), T) = \sum_{r=0}^{\ell+1} C_r(\mathbf{q})T^{\ell+1-r} ,$$

again with $C_0(\mathbb{Q}) = 1$.

To compute the $c_r(\mathbb{Q})$, we use Newton's formulas linking the sums of powers of the roots of a polynomial with the coefficients of that polynomial. That is, we define power series $s_r(\mathbb{Q})$ by

$$s_r(\mathbf{q}) = \sum_{k=0}^{\ell-1} f(\mathbf{q}^{1/\ell}\zeta_\ell^k)^r ,$$

from which we extract the $c_r(\mathbf{q})$ via the recurrence relation

$$c_r(\mathbf{q}) = \frac{-1}{r}\sum_{k=1}^{r} s_k(\mathbf{q})c_{r-k}(\mathbf{q}) .$$

We do the computations in this way to avoid having to work with power series with fractional powers of $\mathbf{q}$. To see this, we first write the power series of $f(\mathbb{Q})^r$ as

$$f(\mathbf{q})^r = \sum_{n=rv}^{\infty} a_{n,r}\mathbf{q}^n .$$

We then have

$$\begin{aligned} s_r(\mathbf{q}) &= \sum_{k=0}^{\ell-1} f(\mathbf{q}^{1/\ell}\zeta_\ell^k)^r \\ &= \sum_{k=0}^{\ell-1}\sum_{n=rv}^{\infty} a_{n,r}(\mathbf{q}^{1/\ell}\zeta_\ell^k)^n \\ &= \sum_{n=rv}^{\infty} a_{n,r}\mathbf{q}^{n/\ell}\sum_{k=0}^{\ell-1}\zeta_\ell^{kn} \\ &= \sum_{i=\lceil rv/\ell\rceil}^{\infty} (\ell a_{i\ell,r})\,\mathbf{q}^i \end{aligned}$$

since

$$\sum_{k=0}^{\ell-1}\zeta_\ell^{kn} = \begin{cases} 0 & \text{if } \gcd(n,\ell) = 1 \\ \ell & \text{if } \ell \text{ divides } n \end{cases}$$

We can therefore extract the series for $s_r(\mathbf{q})$ from the series for $f(\mathbb{Q})^r$ without doing any of the computations with fractional powers of $\mathbf{q}$.

Once the $c_r(\mathbf{q})$ are known, computing the $C_r(\mathbf{q})$ is relatively easy since

$$\sum_{r=0}^{\ell+1} C_r(\mathbf{q})T^{\ell+1-r} = \Phi(j(\mathbf{q}),T) = (T - f^*(\mathbf{q}))\left(\sum_{r=0}^{\ell} c_r(\mathbf{q})T^{\ell-r}\right) .$$

We then have

$$C_r(\mathbf{q}) = c_r(\mathbf{q}) - f^*(\mathbf{q})c_{r-1}(\mathbf{q})$$

except for the two extreme terms which are

$$C_0(\mathbf{q}) = 1 \qquad \text{and} \qquad C_{\ell+1}(\mathbf{q}) = -f^*(\mathbf{q})c_\ell(\mathbf{q}) .$$

For the computations, we prefer to avoid negative powers of $\mathbf{q}$ whenever possible so the computations can be done through polynomials arithmetic. For $f(\mathbf{q})$, $s_r(\mathbf{q})$ and $c_r(\mathbf{q})$, this is not a problem, but for $f^*(\mathbf{q})$ and $C_r(\mathbb{Q})$ we have a pole at $\mathbb{Q} = 0$. Since $f^*(\mathbf{q})$ has a pole of order $v$, and $C_r(\mathbf{q})$ has a pole of order $\leq v$ coming from a multiplication by $f^*(\mathbf{q})$, we will replace these two series with

$$\hat{f}^*(\mathbf{q}) = q^v f^*(\mathbf{q}) = \left(\frac{\hat{\eta}(\mathbf{q})}{\hat{\eta}(\mathbf{q}^\ell)}\right)^{2s}$$

and

$$\hat{C}_r(\mathbf{q}) = \mathbf{q}^v C_r(\mathbb{Q}) .$$

The computations of the $\hat{C}_r(\mathbb{Q})$ are then done as

$$\begin{aligned} \hat{C}_r(\mathbf{q}) &= \mathbf{q}^v c_r(\mathbf{q}) - \hat{f}^*(\mathbf{q})c_{r-1}(\mathbf{q}) \\ \hat{C}_0(\mathbf{q}) &= \mathbf{q}^v \\ \hat{C}_{\ell+1}(\mathbf{q}) &= -\hat{f}^*(\mathbf{q})c_\ell(\mathbf{q}) . \end{aligned}$$

To obtain $C_r(j)$, we remove powers of $j(\mathbf{q})$ (in decreasing order) from $C_r(\mathbf{q})$ until it becomes 0. To do this, we obverse that if $C_r(\mathbf{q})$ has a pole of order $n$ at $\mathbf{q} = 0$, then the order of the pole can be reduced by removing a constant multiple of $j(\mathbf{q})^n$ (since $j(\mathbf{q})$ has a pole of order 1 at $\mathbf{q} = 0$). in terms of $\hat{C}_r(\mathbf{q})$ and $\hat{\jmath}(\mathbf{q})$, removing $j(\mathbf{q})^n$ from $C_r(\mathbf{q})$ translate into removing $\mathbf{q}^{v-n}\hat{\jmath}(\mathbf{q})^n$ from $\hat{C}_r(\mathbf{q})$, which we do starting with the lowest degree in $\mathbf{q}$.

For the computations, the power series are replaced by polynomials of finite degree, which is taken as small as possible to reduce computational costs. The polynomials must be chosen such that $C_r(\mathbf{q})$ is computed at least up to the constant term since the theory tells us that $C_r(\mathbf{q})$ is a polynomial (of degree at most $v$) in $j(\mathbf{q})$. $\hat{C}_r(\mathbf{q})$ is therefore computed up to the coefficient of $\mathbf{q}^v$. If we want to verify that the implementation is done correctly, we can also compute the series to a higher degree and verify that the remainder of $C_r(\mathbf{q}) - C_r(j(\mathbf{q}))$ is indeed 0. For this we introduce a safety parameter $m$, which is set to zero if we want to optimize speed, but can be increased for testing and safety purposes.

The degrees of the polynomials are given in Table 1. These values are chosen such that $C_r(\mathbf{q}) - C_r(j(\mathbf{q}))$ is computed exactly up to degree $m$ in $\mathbf{q}$. During the computations, the terms of higher degree can then be removed, i.e. if $p(\mathbf{q})$ is computed up to degree $n$, then the computations are done modulo $\mathbf{q}^{n+1}$. Note that a similar approach can be used in the computation of $\Phi_\ell(J,T)$ (previous subsection), where $\hat{\jmath}(\mathbf{q})$ and $\hat{\jmath}(\mathbf{q}^\ell)$ would be needed up to degree $\ell(\ell+1)+m$.

The computation of $\Phi(J,T)$ is summarized in Algorithm 4.

**Algorithm 4** Computing $\Phi(J,T)$

**Inputs:** series $\hat{\jmath}(\mathbf{q})$ and $\hat{\eta}(\mathbf{q})$, prime $\ell$ and parameter $m$.
**Outputs:** The polynomial $\Phi(J,T)$.

$u \leftarrow 12/\gcd(12,\ell-1)$
$v \leftarrow (\ell-1)/\gcd(12,\ell-1)$
$\hat{\eta}_1 = \sum_{i=0}^{\ell(v-1+m)} b_i\mathbf{q}^i \leftarrow \hat{\eta}(\mathbf{q}) \bmod \mathbf{q}^{\ell(v-1+m)+1}$
$\hat{\eta}_\ell \leftarrow \sum_{i=0}^{v-1+m} b_i\mathbf{q}^{i\ell}$
$f^* \leftarrow \hat{\eta}_1/\hat{\eta}_\ell \bmod \mathbf{q}^{\ell(v-1+m)+1}$ (long division)
$\sum_{n=0}^{\ell(v+m)} a_{n,1}\mathbf{q}^n = f_1 \leftarrow \ell^u\mathbf{q}^v/f^* \bmod \mathbf{q}^{\ell(v+m)+1}$ (long division)
$s_1 \leftarrow \sum_{i=0}^{v+m} (\ell a_{i\ell,1})\,\mathbf{q}^i$
**for** $r=2$ to $\ell$ **do**
  $f_r = \sum_{n=0}^{\ell(v+m)} a_{n,r}\mathbf{q}^n \leftarrow f_{r-1}\cdot f_1 \bmod \mathbf{q}^{\ell(v+m)+1}$
  $s_r \leftarrow \sum_{i=0}^{v+m} (\ell a_{i\ell,r})\,\mathbf{q}^i$
**end for**
$c_0 \leftarrow 1$
**for** $r=1$ to $\ell$ **do**
  $c_r \leftarrow \frac{-1}{r}\sum_{k=1}^{r} s_k c_{r-k} \bmod \mathbf{q}^{v+m+1}$
**end for**
$\hat{f}^* \leftarrow \hat{f}^* \bmod \mathbf{q}^{v+m+1}$
**for** $r=1$ to $\ell$ **do**
  $\hat{C}_r \leftarrow \mathbf{q}^v c_r - \hat{f}^* c_{r-1} \bmod \mathbf{q}^{v+m+1}$
**end for**
$\hat{C}_{\ell+1} \leftarrow -\hat{f}^* c_\ell \bmod \mathbf{q}^{v+m+1}$
$\hat{\jmath}_1 \leftarrow \hat{\jmath} \bmod \mathbb{Q}^{v+m+1}$
**for** $i=2$ to $v$ **do**
  $\hat{\jmath}_i \leftarrow \hat{\jmath}_{i-1}\hat{\jmath}_1 \bmod \mathbf{q}^{v+m+1}$
**end for**
**for** $i=1$ to $v-1$ **do**
  $\hat{\jmath}_i \leftarrow \mathbf{q}^{v-i}\hat{\jmath}_i \bmod \mathbf{q}^{v+m+1}$ (correction for the factor $\mathbf{q}^v$)
**end for**
$\hat{\jmath}_0 \leftarrow \mathbf{q}^v$ (constant term)
$C_0(J) \leftarrow 1$
**for** $r=1$ to $\ell+1$ **do**
  **for** $i=v$ to $0$ **do**
    $C_{r,i} \leftarrow$ coefficient of $\mathbf{q}^{v-i}$ in $\hat{C}_r$
    $\hat{C}_r \leftarrow \hat{C}_r - C_{r,i}\hat{\jmath}_i$
  **end for**
  **if** $\hat{C}_r \neq 0$ **then**
    Return error message
  **end if**
  $C_r(J) \leftarrow \sum_{i=0}^{v} C_{r,i}J^i$
**end for**
**Return** $\Phi(J,T) = \sum_{r=0}^{\ell+1} C_r(J)T^{\ell+1-r}$

**Table 1** Degrees in $\mathbf{q}$

| polynomial | degree |
|---|---|
| $\hat{\eta}(\mathbf{q})$ | $\ell(v-1+m)$ |
| $\hat{f}^*(\mathbf{q})$ | $\ell(v-1+m)$ |
| $f(\mathbf{q})$ | $\ell(v+m)$ |
| $f(\mathbf{q})^r$ | $\ell(v+m)$ |
| $s_r(\mathbf{q})$ | $v+m$ |
| $c_r(\mathbf{q})$ | $v+m$ |
| $\hat{C}_r(\mathbf{q})$ | $v+m$ |
| $\hat{\jmath}(\mathbf{q})$ | $v+m$ |

## 2.7 Computing $p_1$, $\tilde{a}$ and $\tilde{b}$

In this section and Section 2.8 we will see how to obtain a factor of the division polynomial $f_\ell(X)$ when $\ell$ is an Elkies prime for the curve $E$. We will first use relations based on the polynomial $\Phi_\ell(J,T)$ or its equivalents ($\Phi(J,T)$ and $\Phi_\ell^*(J,T)$) to compute the equation of the isogenous curve $\tilde{E} : y^2 = x^3 + \tilde{a}x + \tilde{b}$. This is based on the following proposition linking elliptic curves and modular forms:

**Proposition 2.37** *For each value of* $\mathbf{q}$*, we have an elliptic curve* $E$ *given by the equation*

$$y^2 = x^3 - \frac{E_4(\mathbf{q})}{48}x + \frac{E_6(\mathbf{q})}{864}$$

*and with* $j$*-invariant* $j(\mathbf{q})$.

At the same time, we will obtain preliminary information on the points in the kernel of the isogeny. We define $p_1$ as the sum of the roots of the factor of $f_\ell(X)$ corresponding to the kernel of the isogeny, i.e. $\frac{1}{2}$ of the sum of the $x$ coordinates of the points (other than $P_\infty$) in the kernel of the isogeny. To do this, we use a result linking the roots of the isogeny with the modular form $E_2$ defined by

$$E_2(\mathbf{q}) = 1 - 24\sum_{n=1}^{\infty}\frac{n\mathbf{q}^n}{1-\mathbf{q}^n} \quad .$$

**Proposition 2.38** *Given the isogenous curves* $E$ *and* $\tilde{E}$ *corresponding to* $\mathbf{q}$ *and* $\mathbf{q}^\ell$ *respectively, then*

$$p_1 = \frac{\ell}{24}\left(E_2(\mathbf{q}) - \ell E_2(\mathbf{q}^\ell)\right) \quad .$$

In Section 2.8 we will see that we can obtain the factor of $f_\ell(X)$ from $p_1$ and the equation of the curves $E$ and $\tilde{E}$.

Since the polynomials $\Phi_\ell(J,T)$, $\Phi(J,T)$ and $\Phi_\ell^*(J,T)$ are based on relations with different modular functions, the method used to extract information on the isogenous curve will be different. However, the three methods make use of a common set of results.

First of all, we recall that the modular forms are written in terms of $\mathbf{q}$, although they are really functions of $\tau$ (which is in the complex upper half-plane) with $\mathbf{q} = \exp(2\pi i\tau)$. This means that for a modular form $M(\mathbf{q})$, the derivative is really

in terms of $\tau$ and we have

$$M'(\mathbf{q}) = \frac{dM(\mathbf{q})}{d\tau} = \frac{dM(\mathbf{q})}{d\mathbf{q}} \cdot \frac{d\mathbf{q}}{d\tau} \ .$$

To simplify the constants, we treat $2\pi i\tau$ as the variable instead of $\tau$, which gives us

$$M'(\mathbf{q}) = \left.\frac{dM(x)}{dx}\right|_{x=\mathbf{q}} \cdot \mathbf{q}$$

where $M(x)$ is $M(\mathbf{q})$ in terms of the variable $\mathbf{q}$ (instead of $2\pi i\tau$). To simplify the notation, we will usually write $M(\mathbf{q})$ as $M$, understanding that the modular form relates to the curve $E$ associated to $\mathbf{q}$. Since we will also use modular forms related to the isogenous curve $\tilde{E}$ associated to $\mathbf{q}^\ell$, we will denote these modular forms with the symbol $\tilde{}$, i.e. $\tilde{M} = M(\mathbf{q}^\ell)$.

We first have Theorem 2.39, due to Ramanujan, which links the modular forms $E_2$, $E_4$ and $E_6$ and their derivatives, from which we can easily obtain Proposition 2.40 which gives us relations with the derivatives of $j(\mathbb{Q})$.

**Theorem 2.39** *The derivatives of the modular forms $E_2$, $E_4$ and $E_6$ satisfy*

$$E_2' = \frac{1}{12}\left(E_2^2 - E_4\right) \tag{2.13}$$

$$E_4' = \frac{1}{3}\left(E_2 E_4 - E_6\right) \tag{2.14}$$

$$E_6' = \frac{1}{2}\left(E_2 E_6 - E_4^2\right) \ . \tag{2.15}$$

**Proposition 2.40** *The first two derivatives of the modular form $j$ satisfy*

$$j' = -j\frac{E_6}{E_4} \tag{2.16}$$

$$j' = -(j-1728)\frac{E_4^2}{E_6} \tag{2.17}$$

$$j'' = j'\left(\frac{1}{6}E_2 - \frac{1}{2}\frac{E_4^2}{E_6} - \frac{2}{3}\frac{E_6}{E_4}\right) \ . \tag{2.18}$$

In terms of the curve $\tilde{E}$ (associated to $\mathbf{q}^\ell$), we have

$$\tilde{M}' = \left(M(\mathbf{q}^\ell)\right)' = \left.\frac{dM(x)}{dx}\right|_{x=\mathbf{q}^\ell} \cdot \ell\mathbf{q}^\ell$$

and Theorem 2.39 and Proposition 2.40 give

$$\tilde{E}_2' = \frac{\ell}{12}\left(\tilde{E}_2^2 - \tilde{E}_4\right) \tag{2.19}$$

$$\tilde{E}_4' = \frac{\ell}{3}\left(\tilde{E}_2\tilde{E}_4 - \tilde{E}_6\right) \tag{2.20}$$

$$\tilde{E}_6' = \frac{\ell}{2}\left(\tilde{E}_2\tilde{E}_6 - \tilde{E}_4^2\right) \tag{2.21}$$

$$\tilde{\jmath}' = -\ell\tilde{\jmath}\frac{\tilde{E}_6}{\tilde{E}_4} \tag{2.22}$$

$$\tilde{\jmath}' = -\ell(\tilde{\jmath} - 1728)\frac{\tilde{E}_4^2}{\tilde{E}_6} \tag{2.23}$$

$$\tilde{\jmath}'' = \ell\tilde{\jmath}'\left(\frac{1}{6}\tilde{E}_2 - \frac{1}{2}\frac{\tilde{E}_4^2}{\tilde{E}_6} - \frac{2}{3}\frac{\tilde{E}_6}{\tilde{E}_4}\right) . \tag{2.24}$$

**2.7.1 From $\Phi_\ell(J,T)$.** The modular equation $\Phi_\ell(J,T)$ links the $j$-invariant of the curve $E$ to the $j$-invariant of the isogenous curve $\tilde{E}$, so it should be no surprise that we will use it to obtain information on the isogenous curve. To do this, we will take advantage of the derivatives of $\Phi_\ell(J,T)$ and the link they establish between the roots. Although the results are established in terms of the modular forms (i.e. as functions of $\mathbf{q}$), we will only use known values of the modular forms, starting from $E_4$, $E_6$ and $j$ (obtained from Proposition 2.37) and $\tilde{\jmath}$ (an $\mathbb{F}_q$-rational root of $\Phi_\ell(j,T)$), without ever computing $\mathbf{q}$.

We first note that by construction, $\Phi_\ell(j(\mathbf{q}), j(\mathbf{q}^\ell))$ is identically 0, so all its derivatives must also be 0. We can then write

$$0 = (\Phi_\ell(j,\tilde{\jmath}))' = \left(\frac{d}{dJ}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right) j' + \left(\frac{d}{dT}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right)\tilde{\jmath}'$$

and

$$\begin{aligned}
0 &= (\Phi_\ell(j,\tilde{\jmath}))'' \\
&= \left(\frac{d}{dJ}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right) j'' + \left(\frac{d^2}{dJ^2}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right)(j')^2 \\
&\quad + 2\left(\frac{d^2}{dJdT}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right) j'\tilde{\jmath}' \\
&\quad + \left(\frac{d^2}{dT^2}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right)(\tilde{\jmath}')^2 + \left(\frac{d}{dT}\Phi_\ell(J,T)\bigg|_{(J,T)=(j,\tilde{\jmath})}\right)\tilde{\jmath}''
\end{aligned}$$

To simplify the notation, we use the following substitution:

$$D_J = \frac{d}{dJ}\Phi_\ell(J,T)\Big|_{(J,T)=(j,\tilde{\jmath})}$$
$$D_T = \frac{d}{dT}\Phi_\ell(J,T)\Big|_{(J,T)=(j,\tilde{\jmath})}$$
$$D_{JJ} = \frac{d^2}{dJ^2}\Phi_\ell(J,T)\Big|_{(J,T)=(j,\tilde{\jmath})}$$
$$D_{JT} = \frac{d^2}{dJdT}\Phi_\ell(J,T)\Big|_{(J,T)=(j,\tilde{\jmath})}$$
$$D_{TT} = \frac{d^2}{dT^2}\Phi_\ell(J,T)\Big|_{(J,T)=(j,\tilde{\jmath})} .$$

Since $j$ and $\tilde{\jmath}$ are known, the value of these modular forms is easily computed. Note that we will use the same notation in the next two section, substituting $\Phi_\ell(J,T)$ by $\Phi(J,T)$ or $\Phi_\ell^*(J,T)$ and replacing $\tilde{\jmath}$ with $f$.

Using Equation 2.16, we can compute $j'$ from $E_4$, $E_6$ and $j$. We then use the identity $D_J j' + D_T \tilde{\jmath}' = 0$ to obtain $\tilde{\jmath}'$.

We then combine Equations 2.22 and 2.23 to get

$$(\tilde{\jmath}')^2 = \ell^2\tilde{\jmath}(\tilde{\jmath}-1728)\frac{\tilde{E}_6}{\tilde{E}_4}\frac{\tilde{E}_4^2}{\tilde{E}_6} = \ell^2\tilde{\jmath}(\tilde{\jmath}-1728)\tilde{E}_4 \tag{2.25}$$

which allows us to compute $\tilde{E}_4$ and we then use one of the two equations again to find $\tilde{E}_6$. Using Proposition 2.37 on $\tilde{E}_4$ and $\tilde{E}_6$ gives us the equation of the isogenous curve.

To compute $p_1$, we look at the second derivative of $\Phi_\ell(j,\tilde{\jmath})$, which we can now write as $D_J j'' + D_{JJ}(j')^2 + 2D_{JT}j'\tilde{\jmath}' + D_{TT}(\tilde{\jmath}')^2 + D_T\tilde{\jmath}'' = 0$. Using $D_T\tilde{\jmath}' = -D_J j'$ (first derivative), we can rewrite the equality given by the second derivative as

$$\frac{j''}{j'} - \frac{\tilde{\jmath}''}{\tilde{\jmath}'} = -\frac{D_{JJ}(j')^2 + 2D_{JT}j'\tilde{\jmath}' + D_{TT}(\tilde{\jmath}')^2}{D_J j'} .$$

The right-hand side of this equation is known, so we look at $\frac{j''}{j'} - \frac{\tilde{\jmath}''}{\tilde{\jmath}'}$. Using Equations 2.18 and 2.24, we have

$$\begin{aligned}
\frac{j''}{j'} - \frac{\tilde{\jmath}''}{\tilde{\jmath}'} &= \left(\frac{1}{6}E_2 - \frac{1}{2}\frac{E_4^2}{E_6} - \frac{2}{3}\frac{E_6}{E_4}\right) - \ell\left(\frac{1}{6}\tilde{E}_2 - \frac{1}{2}\frac{\tilde{E}_4^2}{\tilde{E}_6} - \frac{2}{3}\frac{\tilde{E}_6}{\tilde{E}_4}\right)\\
&= \frac{1}{6}\left(E_2 - \ell\tilde{E}_2\right) - \frac{1}{2}\left(\frac{E_4^2}{E_6} - \ell\frac{\tilde{E}_4^2}{\tilde{E}_6}\right) - \frac{2}{3}\left(\frac{E_6}{E_4} - \ell\frac{\tilde{E}_6}{\tilde{E}_4}\right)\\
&= \frac{4}{\ell}p_1 - \frac{1}{2}\left(\frac{E_4^2}{E_6} - \ell\frac{\tilde{E}_4^2}{\tilde{E}_6}\right) - \frac{2}{3}\left(\frac{E_6}{E_4} - \ell\frac{\tilde{E}_6}{\tilde{E}_4}\right)
\end{aligned}$$

(using Proposition 2.38), and since we know $E_4$, $E_6$, $\tilde{E}_4$ and $\tilde{E}_6$ we can use this to compute $p_1$.

The computations are summarized in Algorithm 5. Finally, we note that in the unlikely case when $j'$ or $D_J$ are zero (where the arguments presented here cannot be applied since it would imply a division by zero), the order of the curve can be

computed very quickly using a different approach than Schoof's (see [75], end of Section 7, for the details).

---

**Algorithm 5** Computing the isogenous curve from $\Phi_\ell(J,T)$

**Inputs:** the curve parameters $a$ and $b$, the $j$-invariant $j$, the polynomial $\Phi_\ell(J,T)$ and a root $\tilde{\jmath}$ of $\Phi_\ell(j,T)$.

**Outputs:** $\tilde{a}$, $\tilde{b}$ and $p_1$.

Compute the first and second derivatives of $\Phi_\ell(J,T)$ relative to $J$ and $T$

Evaluate the derivatives of $\Phi_\ell(J,T)$ at $(J,T) = (j,\tilde{\jmath})$ to obtain

$D_J$, $D_F$, $D_{JJ}$, $D_{JT}$ and $D_{TT}$.

$E_4 \leftarrow -48a$ and $E_6 \leftarrow 864b$

$j' \leftarrow -\frac{jE_6}{E_4}$

$\tilde{\jmath}' \leftarrow -\frac{D_J j'}{D_T}$

$\tilde{E}_4 \leftarrow \frac{1}{\ell^2}\frac{(\tilde{\jmath}')^2}{\tilde{\jmath}(\tilde{\jmath}-1728)}$

$\tilde{E}_6 \leftarrow -\frac{1}{\ell}\frac{\tilde{E}_4\tilde{\jmath}'}{\tilde{\jmath}}$

$p_1 \leftarrow \frac{\ell}{4}\left(\frac{1}{2}\left(\frac{E_4^2}{E_6} - \ell\frac{\tilde{E}_4^2}{\tilde{E}_6}\right) - \frac{2}{3}\left(\frac{E_6}{E_4} - \ell\frac{\tilde{E}_6}{\tilde{E}_4}\right) - \frac{D_{JJ}(j')^2 + 2D_{JT}j'\tilde{\jmath}' + D_{TT}(\tilde{\jmath}')^2}{D_J j'}\right)$

$\tilde{a} \leftarrow -\frac{1}{48}\tilde{E}_4$ and $\tilde{b} \leftarrow \frac{1}{864}\tilde{E}_6$

---

**2.7.2 From $\Phi(J,T)$.** If we are using $\Phi(J,T)$ instead of $\Phi_\ell(J,T)$, then the roots of $\Phi(j,T)$ do not give us $\tilde{\jmath}$, i.e. the obvious link between $E$ and $\tilde{E}$ is lost. However, the choice of the function $f(\mathbb{Q})$ was designed to do just that. By definition, $f = \ell^s(\tilde{\eta}/\eta)^{2s}$, where $s$ divides 12, so

$$f^{12/s} = \ell^{12}\frac{\tilde{\Delta}}{\Delta} ,$$

which gives us a connection between $E$ and $\tilde{E}$ (recall that $j = E_4^3/\Delta$).

Let $f$ be an $\mathbb{F}_q$-rational root of $\Phi(j,T)$. Using the same notation for the derivatives of $\Phi(J,T)$ evaluated at $(J,T) = (j,f)$ as we did in the case of $\Phi_\ell$, we have

$$D_J j' + D_T f' = 0 \tag{2.26}$$

and

$$D_J j'' + D_{JJ}(j')^2 + 2D_{JT}j'f' + D_{TT}(f')^2 + D_T f'' = 0 . \tag{2.27}$$

If we differentiate $f$, we find

$$\begin{aligned} f' &= \left(\ell^s\left(\frac{\tilde{\eta}}{\eta}\right)^{2s}\right)' \\ &= 2s\left(\frac{\tilde{\eta}'}{\tilde{\eta}} - \frac{\eta'}{\eta}\right)f . \end{aligned}$$

If we differentiate $\eta(\mathbb{Q})$, we find

$$\begin{aligned}
\eta' &= \left(\mathbf{q}^{1/24}\prod_{n=1}^{\infty}(1-\mathbf{q}^n)\right)' \\
&= \left(\frac{1}{24}\mathbf{q}^{-23/24}\prod_{n=1}^{\infty}(1-\mathbf{q}^n)+\mathbf{q}^{1/24}\sum_{n=1}^{\infty}\left((-n)\mathbf{q}^{n-1}\prod_{i\neq n}(1-\mathbf{q}^i)\right)\right)\mathbf{q} \\
&= \eta(\mathbf{q})\left(\frac{1}{24}-\sum_{n=1}^{\infty}\frac{n\mathbf{q}^n}{(1-\mathbf{q}^n)}\right) \\
&= \frac{1}{24}\eta E_2 \ ,
\end{aligned}$$

and similarly for $\tilde{\eta}$ we have $\tilde{\eta}' = \frac{\ell}{24}\tilde{\eta}\tilde{E}_2$. Combining these into $f'$, we find

$$f' = \frac{s}{12}\left(\ell\tilde{E}_2 - E_2\right)f.$$

To simplify the notation, we define $Z$ as $Z = f'/f$, and Proposition 2.38 becomes

$$p_1 = -\frac{\ell}{2s}Z \ .$$

Since $j'$ can be obtained from Equation 2.16, Equation 2.26 gives us $f'$, which in turn gives us $Z$ and $p_1$.

We now consider the derivative of $Z$. Since $Z = \frac{f'}{f}$, we have

$$\begin{aligned}
Z' &= \frac{f''}{f} - \left(\frac{f'}{f}\right)^2 \\
&= -\frac{D_J j'' + D_{JJ}(j')^2 + 2D_{JT}j'f' + D_{TT}(f')^2}{fD_T} - Z^2
\end{aligned}$$

(using Equation 2.27 to substitute $f''$), and combining with Equations 2.27 and 2.18, we get

$$\begin{aligned}
Z' &= -\frac{D_J}{D_T}\frac{j''}{f} - \frac{D_{JJ}(j')^2 + 2D_{JT}j'f' + D_{TT}(f')^2}{fD_T} - Z^2\,() \\
&= \frac{f'}{j'}\frac{j''}{f} - \frac{D_{JJ}(j')^2 + 2D_{JT}j'f' + D_{TT}(f')^2}{fD_T} - Z^2 \\
&= Z\left(\frac{1}{6}E_2 - \frac{1}{2}\frac{E_4^2}{E_6} - \frac{2}{3}\frac{E_6}{E_4}\right) - \frac{D_{JJ}(j')^2+2D_{JT}j'f'+D_{TT}(f')^2}{fD_T} - Z^2 \ . \qquad (2.28)
\end{aligned}$$

On the other hand, we showed that $Z = \frac{s}{12}\left(\ell\tilde{E}_2 - E_2\right)$, so we also have (using Equations 2.13 and 2.19)

$$\begin{aligned}
Z' &= \frac{s}{12}\left(\ell\tilde{E}_2' - E_2'\right) \\
&= \frac{s}{12}\left(\frac{\ell^2}{12}\left(\tilde{E}_2^2 - \tilde{E}_4\right) - \frac{1}{12}\left(E_2^2 - E_4\right)\right) \\
&= \frac{s}{144}\left(\ell^2\tilde{E}_2^2 - E_2^2\right) - \frac{s\ell^2}{144}\tilde{E}_4 + \frac{s}{144}E_4 \ . \qquad (2.29)
\end{aligned}$$

Since $Z^2 = \frac{s^2}{144}\left(\ell^2 \tilde{E}_2^2 - 2\ell \tilde{E}_2 E_2 + E_2^2\right)$, the Equation 2.29 becomes

$$\begin{aligned} Z' &= \frac{1}{s}Z^2 + \frac{s}{144}\left(2\ell \tilde{E}_2 E_2 - 2E_2^2\right) - \frac{s\ell^2}{144}\tilde{E}_4 + \frac{s}{144}E_4 \\ &= \frac{1}{s}Z^2 + \frac{1}{6}ZE_2 - \frac{s\ell^2}{144}\tilde{E}_4 + \frac{s}{144}E_4 \ . \end{aligned} \tag{2.30}$$

We can now take the difference between Equations 2.28 and 2.30, which cancels the terms $\frac{1}{6}ZE_2$, allowing us to compute $\tilde{E}_4$ (since the values of the other expressions are known). Once $\tilde{E}_4$ is known, we combine it with $\tilde{\Delta}$ to get $\tilde{\jmath}$.

To compute $\tilde{E}_6$, we will use Equation 2.22, but for this we need $\tilde{\jmath}'$. In order to find $\tilde{\jmath}$, we will look at the first derivative of $\Phi(j^*, f^*) = \Phi(\tilde{\jmath}, f^*)$ (recall that $j^* = \tilde{\jmath}$). Since $\Phi(j, f) = 0$ for every value of $q$, $\Phi(j^*, f^*)$ will also be 0 and so will its derivatives. Denoting $D_J^*$ and $D_T^*$ the partial derivatives of $\Phi(J, T)$ relative to $J$ and $T$ evaluated at $(J, T) = (\tilde{\jmath}, f^*)$, we have an equivalent relation to Equation 2.26:

$$D_J^* \tilde{\jmath}' + D_T^* (f^*)' = 0 \ .$$

Furthermore, we have

$$(f^*)' = \left(\ell^s \frac{1}{f}\right)' = -\ell^s \frac{f'}{f^2} = -Zf^* \ ,$$

which allows us to compute $\tilde{\jmath}'$ and $\tilde{E}_6$.

The computations are summarized in Algorithm 6.

## 2.8 Computing the factor of $f_\ell$

Once we have the equation of the isogenous curve and the sum of the $x$ coordinates of the points in the kernel of the isogeny, we can compute a factor $g_\ell(X)$ of degree $(\ell - 1)/2$ of the division polynomial $f_\ell(X)$.

We begin by establishing the notation. Given the elliptic curve $E$ defined by $y^2 = x^3 + ax + b$ and the isogenous curves $\tilde{E}$ defined by $y^2 = x^3 + \tilde{a}x + \tilde{b}$, we have the Weierstraß $\mathcal{P}$-functions

$$\mathcal{P}(z) = \frac{1}{z^2} + \sum_{n=1}^{\infty} c_n z^{2n}$$

and

$$\tilde{\mathbb{P}}(z) = \frac{1}{z^2} + \sum_{n=1}^{\infty} \tilde{c}_n z^{2n} \ .$$

For $E$, we have $c_1 = -a/5$, $c_2 = -b/7$ and the remaining $c_k$ are given by:

$$c_k = \frac{3}{(k-2)(2k+3)} \sum_{i=1}^{k-2} c_i c_{k-1-i} \ . \tag{2.31}$$

Similarly, for the isogenous curve we have $\tilde{c}_1 = -\tilde{a}/5$, $\tilde{c}_2 = -\tilde{b}/7$ and the remaining $\tilde{c}_k$ are given by:

$$\tilde{c}_k = \frac{3}{(k-2)(2k+3)} \sum_{i=1}^{k-2} \tilde{c}_i \tilde{c}_{k-1-i} \ . \tag{2.32}$$

**Algorithm 6** Computing the isogenous curve from $\Phi(J,T)$

**Inputs:** the curve parameters $a$ and $b$, the $j$-invariant $j$, the polynomial $\Phi(J,T)$ and a root $f$ of $\Phi(j,T)$.

**Outputs:** $\tilde{a}$, $\tilde{b}$ and $p_1$.

Compute the first and second derivatives of $\Phi(J,T)$ relative to $J$ and $T$

Evaluate the derivatives of $\Phi(J,T)$ at $(J,T)=(j,f)$ to obtain

$D_J$, $D_F$, $D_{JJ}$, $D_{JT}$ and $D_{TT}$.

$E_4 \leftarrow -48a$ and $E_6 \leftarrow 864b$

$\Delta = \frac{E_4^3}{j}$

$j' \leftarrow -j\frac{E_6}{E_4}$

$\tilde{\Delta} \leftarrow \Delta\frac{f^{12/s}}{\ell^{12}}$

$f' \leftarrow -\frac{j' D_J}{D_T}$

$Z \leftarrow \frac{f'}{f}$

$\tilde{E}_4 \leftarrow \frac{144}{s\ell^2}\left(Z\left(\frac{1}{2}\frac{E_4^2}{E_6}+\frac{2}{3}\frac{E_6}{E_4}\right)+\frac{(j')^2D_{JJ}+2j'f'D_{JT}+(f')^2D_{TT}}{fD_T}+\frac{s+1}{s}Z^2+\frac{s}{144}E_4\right)$

$\tilde{j} \leftarrow \frac{\tilde{E}_4^3}{\tilde{\Delta}}$

$f^* \leftarrow \ell^s\frac{1}{f}$

Evaluate the derivatives of $\Phi(J,T)$ at $(J,T)=(\tilde{j},f^*)$ to obtain $D_J^*$ and $D_T^*$

$(f^*)' \leftarrow -Zf^*$

$\tilde{j}' \leftarrow -\frac{(f^*)'D_T^*}{D_J^*}$

$\tilde{E}_6 \leftarrow -\frac{1}{\ell}\frac{\tilde{E}_4\tilde{j}'}{\tilde{j}}$

$p_1 \leftarrow -\frac{\ell}{2s}Z$

$\tilde{a} \leftarrow -\frac{1}{48}\tilde{E}_4$ and $\tilde{b} \leftarrow \frac{1}{864}\tilde{E}_6$

We let $g_\ell(X)$ be the polynomial that vanishes at the $x$ coordinates of the points in the kernel of the isogeny $E \to \tilde{E}$. From Proposition 2.34, the roots of $g_\ell(X)$ (in terms of $\mathbb{C}/L$) are of the form

$$\mathcal{P}(r\omega)$$

for some value $\omega$ (which depends on the isogeny), and where $r$ takes the values 1 up to $(\ell-1)/2$. From this we define

$$p_i = \sum_{r=1}^{(l-1)/2} \mathbb{P}(r\omega)^i ,$$

with $p_0 = (\ell-1)/2$ and $p_1$ corresponding to its definition in the previous section.

**2.8.1 Using the derivatives of $\mathcal{P}(z)$:** To compute $g_\ell(X)$, we will take advantage of the derivatives of $\mathcal{P}(z)$. To do this, we will need to write the $2k$-th derivative of $\mathcal{P}(z)$ as a polynomial function of $\mathcal{P}(z)$ of degree $k+1$, i.e. in the form

$$P_k(\mathbb{P}(z)) = \frac{d^{2k}\mathcal{P}(z)}{dz^{2k}} = \sum_{i=0}^{k+1}\mu_k(i)\mathcal{P}(z)^i .$$

Let us first consider how to prove that the $2k$-th derivative of $\mathcal{P}(z)$ is indeed a polynomial function of $\mathcal{P}(z)$ and how we can compute it.

The argument works as a proof by induction, and we first establish the recurrence relation:

$$\begin{aligned}
\frac{d^{2k+2}\mathcal{P}}{dz^{2k+2}} &= \frac{d^2}{dz^2}\left(\frac{d^{2k}\mathcal{P}}{dz^{2k}}\right) \\
&= \frac{d}{dz}\left(\frac{d}{d\mathcal{P}}\left(\frac{d^{2k}\mathcal{P}}{dz^{2k}}\right)\frac{d\mathcal{P}}{dz}\right) \\
&= \left(\frac{d}{d\mathcal{P}}\left(\frac{d^{2k}\mathcal{P}}{dz^{2k}}\right)\frac{d^2\mathcal{P}}{dz^2}\right) + \left(\frac{d^2}{d\mathcal{P}^2}\left(\frac{d^{2k}\mathcal{P}}{dz^{2k}}\right)\left(\frac{d\mathcal{P}}{dz}\right)^2\right) \\
&= \left(6\mathcal{P}^2 + 2a\right)\frac{d}{d\mathcal{P}}\left(\frac{d^{2k}\mathcal{P}}{dz^{2k}}\right) + \left(4\mathcal{P}^3 + 4a\mathcal{P} + 4b\right)\frac{d^2}{d\mathcal{P}^2}\left(\frac{d^{2k}\mathcal{P}}{dz^{2k}}\right)
\end{aligned}$$

(since $\mathcal{P}''(z) = 6\mathcal{P}(z)^2 + 2a$ and $\left(\mathcal{P}'(z)\right)^2 = 4\mathcal{P}(z)^3 + 4a\mathcal{P}(z) + 4b$ from Proposition 2.1). At the same time, $\mathcal{P}''(z) = 6\mathcal{P}(z)^2 + 2a = P_1(\mathcal{P}(z))$, which gives us a starting point to obtain the other $P_k(\mathbb{P}(z))$ by recurrence, i.e. we let

$$P_k(X) = \left.\frac{d^{2k}\mathcal{P}(z)}{dz^{2k}}\right|_{\mathbb{P}(z)=X} = \sum_{i=0}^{k+1} \mu_k(i) X^i$$

and we have

$$\begin{aligned}
P_1(X) &= 6X^2 + 2a \\
P_{k+1}(X) &= P_1(X){P_k}'(X) + 4Y^2{P_k}''(X)
\end{aligned}$$

with $Y^2 = X^3 + aX + b$ as usual.

The link between the derivatives of $\mathcal{P}(z)$ and the roots of $g_\ell(X)$ is found in the following proposition:

**Proposition 2.41** *For every $k \geq 1$, we have*

$$(2k)!(\tilde{c}_k - c_k) = 2(\mu_k(0)p_0 + \ldots + \mu_k(k+1)p_{k+1}).$$

Since $p_0 = (\ell - 1)/2$ and $p_1$ are already known, we can use the values of $c_k$ and $\hat{c}_k$ (obtained using Equations 2.31 and 2.32) to compute $p_k$ starting from $p_2$. From the values of the $p_k$'s we can compute the coefficients of $g_\ell(X)$ using Newton sums. If

$$g_\ell(X) = \sum_{i=0}^{(\ell-1)/2} d_i X^{(\ell-1)/2-i}$$

Then $d_0 = 1$, $d_1 = -p_1$, and we can compute the other $d_i$'s starting from $d_2$ with the recurrence relation

$$-rd_r = \sum_{i=1}^{r} p_i d_{r-i} \ .$$

**2.8.2 Testing the factor of $f_\ell(X)$.** In Section 2.7 (and again in this section), we assumed that the root of $\Phi_\ell(J,T)$ or $\Phi(J,T)$ corresponded to the isogeneous curve associated to $\mathbf{q}^\ell$. However, $\Phi_\ell^*(j,T)$ has in general two roots in $\mathbb{F}_q$ for Elkies primes, but only one of them correspond to the isogeny we decided to use. The second root is in fact "wrong" for our construction, and will lead to invalid factors of $f_\ell(X)$ (the constructed curve may not really be isogenous to $E$). Unfortunately, it is not possible to distinguish the incorrect roots from the correct ones simply from the equations of the isogenous curves. The only practical solution is to try to

compute $g_\ell(X)$ (assuming that we have one of the correct root), and then verify that the polynomial is indeed a factor of $f_\ell(X)$. We can then go through the $\mathbb{F}_q$-rational roots of $\Phi^*_\ell(j,T)$ one after the other until we find a valid factor

As it is often the case in factorization/primality situations, we have two types of tests available to us: a very fast but probabilistic test and a much slower but deterministic test. In general we will run the probabilistic test a number of times until the probability that we are getting "false positives" is small enough that we are almost certain $g_\ell(X)$ must be a factor of $f_\ell(X)$, and then use the deterministic test to remove all doubts.

**Probabilistic test:** In the previous subsection, we computed the values of $p_k$ for $k \le (\ell-1)/2$, however this computation can easily be pushed further. If the $p_k$'s are indeed sums of powers of the roots of $g_\ell(X)$ (i.e. if we really have an isogenous curve $\tilde{E}$), then we must have

$$\begin{aligned}\sum_{j=0}^{(\ell-1)/2} d_j p_{i+j} &= \sum_{j=0}^{(\ell-1)/2}\left(d_j \sum_{r=1}^{(\ell-1)/2} \mathcal{P}(r\omega)^{i+j}\right)\\ &= \sum_{r=1}^{(\ell-1)/2}\sum_{j=0}^{(\ell-1)/2} d_j\mathcal{P}(r\omega)^{i+j}\\ &= \sum_{r=1}^{(\ell-1)/2}\sum_{j=0}^{(\ell-1)/2} g_\ell(\mathcal{P}(r\omega))\mathcal{P}(r\omega)^{i}\\ &= 0\end{aligned}$$

for every $i > 0$ since the roots of $g_\ell(X)$ are $\mathcal{P}(w), \mathcal{P}(2w), \ldots, \mathcal{P}(\frac{\ell-1}{2}w)$. On the other hand, the values of $p_k$ are expected to behave essentially as random elements of $\mathbb{F}_q$ if the computed curve is not isogenous to $E$, so the probability of the sums $\sum_{j=0}^{(\ell-1)/2} d_j p_{i+j}$ being 0 is quite low in that case. We can therefore identify almost all invalid factors (i.e. invalid isogenous curves $\tilde{E}$) by verifying whether or not $\sum_{j=0}^{(\ell-1)/2} d_j p_{i+j}$ is 0 for $i \le 4$.

**Deterministic test:** To show beyond doubt that $g_\ell(X)$ is a factor of $f_\ell(X)$, we have to compute $f_\ell(X) \bmod g_\ell(X)$. We have two possible ways to do this:

1. reduce $f_\ell(X)$ modulo $g_\ell(X)$ is $f_\ell(X)$ is already known, or
2. construct $f_\ell(X) \bmod g_\ell(X)$ using the relations of Propositions 2.14 and 2.17 taken modulo $g_\ell(X)$.

The result will be 0 if and only if $g_\ell(X)$ is a factor of $f_\ell(X)$.

## 2.9 Parallelization

In this section, we describe a parallelized implementation of the SEA algorithm using a server and slaves computing scheme. In the beginning phase, the server sends the prime 5 to the first available slave, the prime 7 to the second available slave, and so on (the primes 2 and 3 are handled differently). After receiving the prime $\ell$, a slave performs the necessary computations and determines whether the prime will be used in the computation of the trace $t$ of the Frobenius map. If the prime is used, the slave then computes $t_\ell \equiv t \bmod \ell$. Once the slave determines that a prime will not be used or finishes computing $t_\ell$, it will be asked to process the

**Table 2** Timings at various field sizes

| Field size in bits | Time in seconds |
|---|---|
| 155 | 8.71 |
| 160 | 9.46 |
| 165 | 10.68 |
| 170 | 11.95 |

next available prime. When $t \bmod \ell$ has been computed for enough small primes, all the slaves are stopped and the server proceeds to compute $t$ using the Chinese Remainder Theorem.

The implementation used the MPICH library (an implementation of the Message Passing Interface 2) and was tested on a cluster of five AMD XP 1700+ processors (one server, four slaves). Computation times for randomly chosen curves over a set of prime field can be found in Table 2. Note that the parallel implementation was not fully optimized, and that some of the techniques described in the previous sections were not used (in particular with the computation of the Frobenius of the polynomials $f_k$).

**2.9.1 Procedure.** We now outline the operation sequences for the server and the slaves. It should be noted that the steps for the two sides are interrelated, and at times one must wait for the other side to finish in order to move on.

For simplicity we will restrict to using only small primes $\ell$, and not prime powers.

We use the following parameters:

- $\ell$: current prime;
- P_l: product of the primes in use, initialized at 1;
- Atkin_max: upper bound for the Atkin primes (from experiments);
- l_max: largest prime for which $\Phi_\ell$ is precomputed.

**Server side:**

1. Send a prime to each of the available slaves (keeping track of the largest prime sent).
2. Wait for a reply.
3. When a slave to return a pair of values (move_on,$t_\ell$):
   - If move_on is $-1$ ($\ell$ is not used), go to step 4.
   - If (move_on,$t_\ell$) is $(\ell, -1)$ ($\ell$ is used, but $t_\ell$ is not yet computed), multiply P_l by $\ell$ and go back to step 2.
   - if (move_on,$t_\ell$) is $(\ell, t_\ell)$ with $t_\ell > 0$ ($t_\ell$ has been computed), record the pair $(\ell, t_\ell)$ and go to step 4.
4. If the following conditions are satisfied:
   - P_l is less than $4\sqrt{q}$ (not enough primes are used to be able to determine $t$), and
   - the next prime $\ell$ is at most l_max ($\Phi_\ell$ is precomputed),

   then send $\ell$ to the slave and return to step 2, otherwise go to step 5.
5. If P_l is less than $4\sqrt{q}$ abort all computations and set P_l to 0 ($t$ cannot be computed under the current settings), otherwise go to step 6.
6. Stop all slaves that are not computing values of $t_\ell$ (there is no need to add more primes) and wait until all the missing pairs $(\ell, t_\ell)$ are recorded.

7. Once all the pairs $(\ell, t_\ell)$ have been returned, use the Chinese Remainder Theorem to compute $t$.

**Slave side:**

1. Receive a prime $\ell$ from the server.
2. Decide if $\ell$ is an Atkin or Elkies prime:
   - If $\ell$ is an Elkies prime, set move_on to $\ell$;
   - If $\ell$ is an Atkin prime no larger than Atkin_max, set move_on to $\ell$, otherwise set move_on to $-1$.
3. Send (move_on,$-1$) to the server.
4. If move_on is $-1$, go back to step 1, otherwise to step 5.
5. Compute $t_\ell$, send (move_on,$t_\ell$) to the server and go back to step 1.

**Acknowledgments.** We would like to thank Zubair Adamjee and Roanne Rondina for their important contribution during the implementation of the algorithm as well as their help in the redaction of Section 2.9.

**Nicolas Thériault**
Instituto de Matemática y Física
Universidad de Talca
Talca, Chile
ntheriau@inst-mat.utalca.cl

Fields Institute Communications
Volume **58**, 2010

CHAPTER 3

# Report on the Denef-Vercauteren/Kedlaya Algorithm

Zubair Ashraf, Ali Juma, and Pramathanath Sastry

**Abstract.** We give a quick survey of the Denef-Vercauteren algorithm—which in turn is an extension of Kedlaya's algorithm to hyperelliptic curves over finite fields of characteristic 2—and report on how we implemented this algorithm at the GANITA Laboratory.

## 3.1 Background

Let $C$ be a smooth complete (i.e. projective) curve of genus $g$ over a finite field $k = \mathbb{F}_q$ and $J = \text{Jac}(C)$ the *Jacobian variety* of $C$. This is a $g$–dimensional projective group variety over $k$ which "parameterizes" the rational equivalence classes of degree zero divisors on $C$ (or equivalently the isomorphism classes of line bundles on $C$ of degree zero). Finding the number of $k$–rational points on $J$ is a problem of considerable importance in cryptography. This report is on our implementation of the work initiated by Kedlaya [48] for $C$ a hyperelliptic curve and $k$ a field of odd characteristic, and extended by Denef and Vercauteren [22], [23] to fields of characteristic 2. In fact, when it comes to specifics, we concentrate only on the work done by Denef and Vercauteren.

The number of $k$ rational points on $J$ is computed by first finding the zeta function $Z(C, t)$ of $C$. In greater detail, recall that

$$Z(C,\, t) := \exp\left(\sum_{r=1}^{\infty} \frac{N_r}{r} t^r\right) \qquad \textbf{(Zeta)}$$

where $N_r = \#$ of $\mathbb{F}_{q^r}$–rational points on $C$. It is well-known, and a part of the Weil Conjectures (proved by Weil himself for curves in 1948) that

$$Z(C,\, t) = \frac{\Psi(t)}{(1-t)(1-qt)} \qquad (3.1)$$

where $\Psi(t)$ is polynomial of degree $2g$ over the rational numbers. If $j_q$ is the number of $k$–rational points on $J$, then again it is well known that

$$j_q = \psi(1). \qquad (3.2)$$

For this—and other reasons—the focus shifts to finding efficient algorithms and programs for computing the zeta function $Z(C, t)$ of $C$. It turns out that hyperelliptic curves are such that $Z(C, t)$ is tractable.

The report is organized as follows: we first give some theoretical background. We move systematically from the conceptual to the concrete. When in the conceptual mode we may well consider situations which are far more general than what

we deal with in practise–the aim being to contemplate, without added clutter, the ideas behind the computational details. The rigid cohomology interpretation we give of Kedlaya's ideas has also been given by Gerkmann in [37] and [38]. Indeed it was while giving a talk at GANITA with this particular slant that we were made aware of Gerkmann's talk at Florida [37]. Subsequently Gerkmann generously provided us portions of his thesis which was at that point still under preparation. The Hodge theoretic (more precisely the Frobenius theoretic) viewpoint touched upon ever so fleetingly was pointed out to us by Ramesh Sreekantan.

## 3.2 Generalities

In this section we paint with broad strokes. The idea is to give a sense of the theoretical underpinnings of the algorithms we report on. We do not go into the details of anything. For the rest of this paper fix a finite field $k = \mathbb{F}_q$. Let $R = W(k)$ be its associated ring of Witt vectors (cf. §§ 3.4.1), and $K$ the quotient field of $R$. Note that $K$ is a field of characteristic zero.

**3.2.1 The zeta function and eigenvalues of the Frobenius endomorphism.** A strategy for computing the zeta function of a smooth projective variety $X$ of dimension $n$ over $k$ (defined exactly as we defined it for the curve $C$ in (**Zeta**)) is this: compute the characteristic polynomials (one for each $i = 0, \ldots, 2\dim X$) $Q_i(t)$ of the Frobenius endomorphism acting on the "$i$–th cohomology" of $X$ with respect to a "good" cohomology theory for $X$, and let $P_i(t) = t^{\deg Q_i} Q_i(1/t)$. Then (a) $P_0(t) = 1 - t$ (b) $P_{2n}(t) = 1 - q^n t$ and (c) the zeta function is given by

$$Z(X, t) = \prod P_i(t)^{(-1)^{i+1}}. \qquad \textbf{(Zeta-P)}$$

Examples of good cohomology theories are the étale cohomology theory developed by A. Grothendieck and M. Artin and the rigid cohomology theory developed by P. Berthelot [8]. When $X$ is *affine* Berthelot's theory reduces to the cohomology theory of Monsky and Washnitzer (cf. [67], [65] and [66])[1]. We will shortly give a definition of the Monsky-Washnitzer cohomology (MW cohomology) for affine varieties over finite fields. Key facts to remember are: if $X$ is defined over $k$ then all its rigid cohomology groups are *finite dimensional* vector spaces over the characteristic zero field $K$ [10] and (since rigid cohomology is a functor taking values in the category of vector spaces over $K$) the Frobenius endomorphism on $X$ lifts naturally to endomorphism on the rigid cohomology groups as $K$–linear maps.

**3.2.2 Reduction to affine varieties; excision.** Berthelot's rigid cohomology is in general not easy to compute. However if the variety in question is affine, we are dealing with MW cohomology, which is potentially more tractable (and, as we shall see, for affine hyperelliptic curves is indeed computable). Since our primary interest is in smooth projective varieties, therefore as a first step, one tries to compare the rigid cohomology of a smooth projective variety $X$ over a finite field $k$ with that of an affine open subvariety $X'$ of $X$. In greater detail, for any variety $U$ over $k$, let $H^i(U)$ denote the $i$th rigid cohomology of $U$. Let $X'$ and $X$ be as above and let $Z$ be the complement of $X'$ in $X$. Let $n = \dim X$, $m = \dim Z$, and $d = n - m$ the codimension of $Z$ in $X$. Berthelot has a notion of "cohomology with

[1] It should be added that Berthelot's theory is a modification of the crystalline cohomology theory of Grothendieck, which in turn was inspired by the affine theory of Monsky and Washnitzer [67], [65] and [66].

supports" for his theory (cf. [9]), and we denote the $i$th cohomology of $X$ with supports in $Z$ by $H^i_Z(X)$. One of the properties which prompted us to describe rigid cohomology as a "good" theory is the existence of an excision exact sequence—an analogue of the usual excision exact sequence for cohomology in classical algebraic topology—which is of the form:

$$\cdots \to H^{i-1}(X') \to H^i_Z(X) \to H^i(X) \to H^i(X') \to H^{i+1}_Z(X) \to \dots$$

The Frobenius acts on all these vector spaces in a compatible way. More precisely let $F_i^X$, $F_i^{X'}$ and $F_i^{X,Z}$ be the action of Frobenius on $H^i(X)$, $H^i(X')$ and $H^i_Z(X)$. Then the following diagram

$$\begin{array}{ccccccc} H^i_Z(X) & \longrightarrow & H^i(X) & \longrightarrow & H^i(X') & \longrightarrow & H^{i+1}_Z(X) \\ \downarrow{\scriptstyle F_i^{X,Z}} & & \downarrow{\scriptstyle F_i^X} & & \downarrow{\scriptstyle F_i^{X'}} & & \downarrow{\scriptstyle F_{i+1}^{X,Z}} \\ H^i_Z(X) & \longrightarrow & H^i(X) & \longrightarrow & H^i(X') & \longrightarrow & H^{i+1}_Z(X) \end{array}$$

commutes. If $Z$ is also *smooth* over $k$ then, it turns out that under our assumptions the eigenvalues of $F_i^{X,Z}$ and $F_i^X$ are algebraic integers with (archimedean) absolute value $q^{i/2}$.

This provides us a way of comparing the rigid cohomologies (together with the Frobenius operator on them) of $X$ and $X'$. In fact if $H^i_Z(X) = 0$ for some $i$ (and as before $Z$ is smooth), then the above diagram says that $H^i(X)$ can be regarded as a subspace of $H^i(X')$ stable under the action of $F_i^{X'}$, and the image $P$ of $H^i(X')$ in $H^{i+1}_Z(X)$ can be regarded as complement of $H^i(X)$ in $H^i(X')$ *stable under* $F_i^{X'}$ so that the operator $F_i^{X'}$ splits as

$$F_i^{X'} = F_i^X \oplus G$$

respecting the splitting $H^i(X') = H^i(X) \oplus P$. The eigenvalues of $F_i^X$ all have absolute values $q^{i/2}$ and the eigenvalues of $G$ (also algebraic numbers) have absolute value $q^{(i+1)/2}$.[2]

If $X$ is a hyperelliptic curve and $Z$ the smooth scheme of points fixed by the hyperelliptic involution, then $X' = X - Z$ is affine. The heart of the Kedlaya algorithm (as well as the Denef-Vercauteren algorithm) is a way of identifying the "pure weight $i$" subquotient (in fact a summand in the cases we're interested in) of $H^i(X')$ using the hyperelliptic involution on $X$.

**3.2.3 Summary.** It is potentially simpler to compute the rigid cohomology of an affine variety since in this case it is the MW cohomology. However, we are interested in zeta functions of smooth projective varieties. One looks for a suitable affine open subvariety $X'$ of $X$ such that the MW cohomology of $X'$ is in practice computable, and such that the resulting excision exact sequence as well as certain special properties of $(X, X')$ allow one to extract the cohomology (as well as the Frobenius endomorphism) of $X$ from that of $X'$. The zeta function of $X$ is then obtained by computing the characteristic polynomials of the Frobenius operators on $H^i(X)$

[2] For those familiar with Hodge theory for complex algebraic varieties, this is analogous to extracting a pure summand of weight $i$ from a mixed Hodge structure.

## 3.3 Main strategy

As we agreed, $k = \mathbb{F}_q$, $R$ the ring of Witt vectors on $k$ and $K$ the quotient field of $R$. By cohomology we mean rigid cohomology unless otherwise stated. Recall that a hyperelliptic curve is a smooth projective curve $C/k$ for which there is a morphism

$$\pi\colon C \to \mathbb{P}^1 = \mathbb{P}^1_k \qquad \textbf{(Hyp)}$$

of degree 2. We are interested in those $C$ (the so called imaginary quadratic curves) which have an affine open subcurve $C^\circ$ which is planar and such that $C^\circ$ has an equation of the form

$$y^2 + \overline{h}(x)y = \overline{f}(x) \qquad \overline{h}, \overline{f} \in k[X],$$

such that $C - C^\circ$ consists of exactly one $k$–rational point. The requirement that $C$ is imaginary quadratic imposes the following conditions on $\overline{f}$ and $\overline{h}$

- $\deg \overline{f} = 2g + 1$,
- $\deg \overline{h} \leq g$ and if the characteristic of $k$ is odd, then $\overline{h} = 0$.

Since $C$ is one dimensional, we have to only deal with three polynomials in the product decomposition (**Zeta-P**) for $Z(C, t)$, viz., $P_0$, $P_1$ and $P_2$. But $P_0(t) = 1-t$ and $P_2(t) = 1 - qt$. Whence the "numerator" $\Psi(t)$ in the formula (3.1) is $P_1(t)$. In other words if $\chi(t) = \chi_C(t)$ is characteristic polynomial of the $K$–linear $F_1^C$ on $H^1(C)$ then

$$\Psi(t) = t^{2g}\chi(1/t). \qquad (3.3)$$

The polynomial $\chi(t)$ allows us to compute $Z(C, t)$ as well as $j_q$ via the formulae in (3.1) and (3.2). Our focus consequently shifts to computing $\chi(t)$. Broadly this involves two steps:

- Describing $H^1(C)$ as a vector space over $K$ (e.g. presenting a natural basis for $H^1(C)$ in terms of the "variables" $x$ and $y$ occurring in the affine equation for $C^\circ$). We point out that $H^1(C)$ is $2g$–dimensional over $K$.
- Describing the action of the Frobenius map $F_1 = F_1^C$ on $H^1(C)$, i.e. describing the action of $F_1$ on the basis elements found in the previous step. This, of course, amounts to writing out the matrix of $F_1$ with respect to this basis.

**3.3.1 Excision again.** The first of the two steps mentioned above needs as an input the excision strategy we outlined earlier for general smooth projective varieties. The hyperelliptic curve $C$ has a an involution $\iota\colon C \to C$ such that $\pi \circ \iota = \pi$ where $\pi$ is as in (**Hyp**). The "fixed points" of $\iota$ (these may include non $k$-rational points) form a zero dimensional smooth subvariety $Z$ of $C$. Let $C' = C - Z$. Then $C'$ is an affine open subset of the affine plane curve $C^\circ$—indeed the unique $k$–rational point in $C - C^\circ$ is a fixed point of $\iota$. We now proceed with excision with respect to the pair $(C, C')$. General principles[3] imply that $H^1_Z(C) = 0$, whence we have a short exact sequence

$$0 \to H^1(C) \to H^1(C') \to P \to 0 \qquad (3.4)$$

where $P$ is the image of $H^1(C')$ in $H^2_Z(C)$. Topological principles (imagine 1-cycles on a punctured Riemann surface of the kind suggested by $C'$) suggest that $H^1(C)$—thought of as a subspace of $H^1(C')$— is the eigenspace corresponding to the eigenvalue $-1$ of the involution $\iota^*\colon H^1(C') \to H^1(C')$ induced by $\iota$. Kedlaya,

[3]In particular the "Thom isomorphism" $H^i_Z(C) \simeq H^{i-2}(Z)$.

as well as Denef-Vercauteren show that this is indeed so. Note that the Frobenius $F_1^{C'}$ restricts on $H^1(C)$ to $F_i^C$. Moreover, it turns out, that $P$ can be identified with the eigenspace of $\iota^*$ corresponding to the eigenvalue 1. We also mention in the passing that the decomposition

$$H^1(C') = H^1(C) \oplus P \tag{3.5}$$

can be also interpreted as the decomposition of $H^1(C')$ into pure Frobenius structures of weights 1 and 2, i.e. (a) $P$ is stable under $F_2^{C,Z}$ (b) the exact sequence (3.4) is compatible with the various Frobenius maps on the three vector spaces involved (c) $H^1(C)$ is the subspace of $H^1(C')$ on which $F_1^{C'}$ has eigenvalues of absolute value $q^{1/2}$ and (d) $P$ is the subspace of $H^1(C')$ on which $F_1^{C'}$ has eigenvalues of absolute value $q$.

**3.3.2 Summary.** Our task is to find a basis for $H^1(C')$ which reflects the decomposition (3.5)—and hence read off the summand $H^1(C)$—and then write down the action of $F = F_1^C = F_1^{C'}|_{H^1(C)}$ on the part of this basis corresponding to $H^1(C)$, and finally compute the characteristic polynomial $\chi$ of $F$. When this is done we have the formula

$$Z(C, t) = \frac{t^{2g}\chi(1/t)}{(1-t)(1-qt)}. \tag{\textbf{Zeta-X}}$$

## 3.4 Monsky-Washnitzer cohomology

This section is a little varied in tone. We give a few details about elementary matters ($p$–adic numbers, Witt vectors, etc.) but skim over the definition of the dagger ring and of the MW-cohomology referring the reader to [23]. We take the view that we have given a conceptual introduction to the main strategy, and now we need to focus on the algorithm for getting a natural basis for $H^1(C)$ and for the Frobenius matrix on it. Therefore we only expand on matters which seem to us germane to this.

**3.4.1 $p$–adic numbers and Witt vectors.** In this section we describe the ring $R$ and the characteristic 0 field $K$. Let $p$ be the characteristic of the field $k(= \mathbb{F}_q)$ and let $a$ be the positive number satisfying $q = p^a$. Any non-zero rational number $x$ can be written uniquely as

$$x = p^{m_x}\frac{a}{b}$$

where $a$ and $b$ are coprime integers such that $p$ is not a divisor of $a$ or of $b$. If $x = 0$ we set $m_x = \infty$. The *$p$–adic absolute value* of $x$ is then defined to be

$$|x|_p := p^{-m_x}$$

(where $p^{-\infty} := 0$). Recall that the *$p$–adic rationals* $\mathbb{Q}_p$ is the completion of $\mathbb{Q}$ with respect to the $p$–adic absolute value. In explicit terms, $\mathbb{Q}_p$ consists of expansions

$$\sum_{i \geq m} \alpha_i p^i$$

where $m$ is an integer and for each $i$, $0 \leq \alpha_i \leq p-1$. The *$p$–adic integers* $\mathbb{Z}_p$ is the subring of $\mathbb{Q}_p$ consisting of elements $x$ such that $|x|_p \leq 1$, i.e. expansions of the form

$$\sum_{i \geq o} \alpha_i p^i.$$

The ring $\mathbb{Z}_p$ has exactly one maximal ideal (i.e. it is a *local ring*), viz., $p\mathbb{Z}_p$. The quotient field of $\mathbb{Z}_p$ is $\mathbb{Q}_p$ and its residue field is $\mathbb{Z}_p/p\mathbb{Z}_p = \mathbb{F}_p$. Note that $\mathbb{Q}_p$ is a field of characteristic zero. Thus $\mathbb{Z}_p$ "interpolates" between a characteristic $p$ field (viz. $\mathbb{F}_p$) and a characteristic zero field (viz., $\mathbb{Q}_p$) via the relations

$$\mathbb{Q}_p \supset \mathbb{Z}_p \twoheadrightarrow \mathbb{F}_p \qquad \textbf{(Int-p)}$$

There is a similar relationship for every finite extension[4] $k$ of $\mathbb{F}_p$. Suppose $k = \mathbb{F}_q$ where $q = p^a$. There is a unique integral domain $R = W(k)$—the ring of *Witt vectors of* $k$ (cf. [47, 497—500])— such that

- $R$ is of characteristic 0 (i.e. $R$ contains $\mathbb{Z}$).
- $R$ is a local ring whose maximal ideal $\mathfrak{m}_R$ is $pR$, and $R$ is complete in its $p$–adic (i.e. $\mathfrak{m}_R$–adic) topology.
- $R/pR = k$.[5]

In fact $R$ is a noetherian discrete valuation ring with uniformizing parameter $p$. Let $K$ be the quotient field of $R$. Then note that $K$ is of characteristic 0, whence $R$ interpolates between the characteristic $p$ field $k$ and the characteristic 0 field $K$ via the relation

$$K \supset R \twoheadrightarrow k = \mathbb{F}_q \qquad \textbf{(Int-q)}$$

Note that when $k = \mathbb{F}_p$, **(Int-p)** specializes to **(Int-q)**.

We now give an explicit model of $R$ and $K$. Since $\mathbb{F}_p$ is perfect, therefore the extension $\mathbb{F}_p \to \mathbb{F}_q = k$ has a *primitive element* $\theta$, i.e. $\theta$ is such that $k = \mathbb{F}_p[\theta]$, where the irreducible (monic) polynomial $\overline{P}(x) \in \mathbb{F}_p[x]$ of $\theta$ is of degree $a$ (recall $q = p^a$). Let $P(x) \in \mathbb{Z}[x]$ be the monic polynomial with coefficients in $\{0, 1, \ldots, p-1\}$ whose reduction modulo $p$ is $\overline{P}$. Then

$$K := \mathbb{Q}_p[\Theta]. \tag{3.6}$$

where $\Theta$ has minimal polynomial $P$. (To be strictly accurate, we have to fix an algebraic closure $\Omega$ of $\mathbb{Q}_p$, and find $\Theta$ in $\Omega$ satisfying $P$.)

Thus elements of $K$ can be written uniquely as

$$\sum_{k=0}^{a-1} \left( \sum_{i \geq m} \alpha_{ik} p^i \right) \Theta^k$$

with $\alpha_{ik} \in \{0, 1, \ldots, p-1\}$. The arithmetic on expressions as above is the standard one, subject to the rule $P(\Theta) = 0$.

The ring $R$ is then the *integral closure* of $\mathbb{Z}_p$ in $K$, i.e. $R$ consists of elements of $K$ which satisfy a monic polynomial over $\mathbb{Z}_p$ (rather than $\mathbb{Q}_p$). Note that $\Theta \in R$ since $\Theta$ satisfies the monic polynomial $P$. Using this, one concludes that

$$R := \mathbb{Z}_p[\Theta] \tag{3.7}$$

and that elements of $R$ have unique representations of the form

$$\sum_{k=0}^{a-1} \left( \sum_{i=0}^{\infty} \alpha_{ik} p^i \right) \Theta^k,$$

where, as before, $\alpha_{ik} \in \{0, 1, \ldots, p-1\}$.

[4]Or for that matter any extension.

[5]Equality, as usual, means "canonically isomorphic".

**3.4.2 Newton interpolation.** Suppose $V$ is a complete discrete valuation ring with uniformizing parameter $\varpi$. Recall that this means that $V$ has exactly two prime ideals, $(0)$ and $\varpi V$ and $V$ is complete in its $\varpi$–adic topology. Let $L$ be the quotient field of $V$. The classical Newton interpolation technique for finding approximate solutions to implicit equations (which works with caveats in the classical case) holds for the valued field $L$. In greater detail we have the following theorem:

**Theorem 3.1** *Let $f(x) \in V[x]$ be a monic polynomial with coefficients in $V$ and $\alpha_1 \in L$ an element such that $f(\alpha_1) \in \varpi V$ and $f'(\alpha_1)$ is a unit in $V$. Define $\alpha_i$ for $i > 1$ recursively via the relations*

$$\alpha_i = \alpha_{i-1} - \frac{f(\alpha_{i-1})}{f'(\alpha_{i-1})}. \qquad \textbf{(Newt)}$$

*Then the sequence $\{\alpha_n\}$ converges in the $\varpi$–adic topology on $L$ to a root $\alpha \in V$ of $f(x)$. Moreover,*

$$\alpha_i \equiv \alpha_{i-1} \mod \varpi^{i-1}$$

We urge the reader to check that $\alpha_1$—and hence $\alpha_n$ for every $n$—must necessarily lie in $V$, using the fact that $f(\alpha_1) \in V$.

The above is special case of *Hensel's Lemma* for complete local rings. There is a more general form of Newton interpolation involving complete local rings which is logically equivalent to Hensel's Lemma—and there are multivariate versions of this too. We do not need these results, but the interested reader may wish to look at [62], [71], and [86].

**3.4.3 Frobenius on $R$.** Let $\sigma : k \to k$ be the $p$–Frobenius on $k$. Since $0 = \overline{P}(\theta)^p = \overline{P}(\theta^p)$, therefore

$$P(\Theta^p) \in pR.$$

Moreover, the separability of the extension $\mathbb{F}_p \to k$ ensures that $P'(\Theta^p)$ is a unit in $R$. Applying Newton interpolation (to $V = R$) one checks via (**Newt**) that the sequence

$$\Theta_1, \Theta_2, \ldots, \Theta_n, \ldots$$

given by

$$\begin{aligned} \Theta_1 &= \Theta^p \\ &\vdots \\ \Theta_n &= \Theta_{n-1} - \frac{P(\Theta_{n-1})}{P'(\Theta_{n-1})} \\ &\vdots \end{aligned} \qquad (3.8)$$

converges to an element $\Theta^\Sigma \in R$ satisfying the polynomial $P(x)$. This gives us a $\mathbb{Z}_p$–algebra map

$$\begin{aligned} \Sigma : R &\to R \\ \Theta &\mapsto \Theta^\Sigma \end{aligned} \qquad (3.9)$$

and which extends in an obvious way to a map of $\mathbb{Q}_p$ extensions $K \to K$ which we also denote $\Sigma$. We denote the image of an element $\alpha \in K$ under $\Sigma$ by $\alpha^\Sigma$.

The map $\Sigma$ is a "lift" of the Frobenius map $\sigma$ in the following sense: the diagram

$$\begin{array}{ccccc} K & \hookleftarrow & R & \twoheadrightarrow & k \\ \Sigma\downarrow & & \Sigma\downarrow & & \downarrow\sigma \\ K & \hookleftarrow & R & \twoheadrightarrow & k \end{array}$$

commutes, where the arrows in each horizontal row are as in (**Int-q**).

**3.4.4 The dagger construction and MW cohomology.** The fundamental unit in the Monsky-Washnitzer theory is the "dagger construction". Given an affine variety $X$ over $k$, with coordinate ring $\overline{A} = k[x_1, \dots, x_n]/(\overline{f}_1, \dots, \overline{f}_m)$, there is an associated "dagger ring" $A^\dagger$, which is a *flat* $R$–algebra such that $A^\dagger/pA^\dagger = \overline{A}$. The crucial construction here is the dagger ring $D(x_1, \dots, x_n)$ of the polynomial ring $k[x_1, \dots, x_n]$ as the ring of "overconvergent power series" as in formula (2) of [23]. The ring $D(x_1, \dots, x_n)$ is denoted $R < x_1, \dots, x_n >^\dagger$ in [23], and it is not hard to see that polynomials over $R$ are overconvergent, i.e. $R[x_1, \dots, x_n] \subset D(x_1, \dots, x_n)$. It turns out that if $f_i \in R[x_1, \dots, x_n]$ is a lift of $\overline{f}_i$ for $i = 1, \dots, m$[6] such that $R[x_1, \dots, x_n]/(f_1, \dots, f_n)$ is flat over $R$ then

$$A^\dagger = D(x_1, \dots, x_n)/(f_1, \dots, f_m).$$

For further details we refer to the concise introduction to these matters in [23, § 2], especially formula (3) for the definition of overconvergent power series.

The universally finite module of differential 1–forms $A^\dagger$ over $R$ is the $A^\dagger$–module

$$\widetilde{\Omega}^1_{A^\dagger} := (A^\dagger \mathrm{d}x_1 + \cdots + A^\dagger \mathrm{d}x_n)/(\sum_{i=1}^{m} A^\dagger (\sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j})). \tag{3.10}$$

Let $\widetilde{\Omega}^i_{A^\dagger} := \wedge^i \widetilde{\Omega}^1_{A^\dagger}$, and let $d_i : \widetilde{\Omega}^i \to \widetilde{\Omega}^{i+1}$ be the usual exterior product. As in [23, § 2, (5)], $\widetilde{\Omega}^\bullet_{A^\dagger}$ forms a complex with differential $d^\bullet$, and the $i$th MW cohomology of $X$ is

$$H^i(X) := H^i(\widetilde{\Omega}^\bullet_{A^\dagger}) \otimes_R K. \tag{\textbf{MW}}$$

Modulo details of the construction of the dagger ring (and allowing elements of $R$ to act as units) we are essentially looking at "closed" $i$–forms modulo "exact" $i$–forms in a controlled way (taking care of the relative speeds of convergence in the $p$-adic topology and the $(x_1, \dots, x_n)$–adic topology).

### 3.5 Hyperelliptic curves

From now on we assume that the characteristic $p$ of $k = \mathbb{F}_q$ is 2, $q = 2^a$ and that we have a projection $C \xrightarrow{\pi} \mathbb{P}^1$ as in (**Hyp**). Let $C^\circ$ and $C'$ be as in § 3.3. Recall that we wish to (a) compute $H^1(C)$ by splitting $H^1(C')$ into eigenspaces and (b) understand the action of the $q$–power Frobenius $F = F_1^C$ on $H^1(C)$ by understanding (partially at any rate) the action of $F_1^{C'}$ on $H^1(C')$.

---

[6] i.e. $\overline{f}_i$ is the image of $f_i$ under the map $R[x_1, \dots, x_n] \twoheadrightarrow R[x_1, \dots, x_n]/pR[x_1, \dots, x_n] = k[x_1, \dots, x_n]$.

**3.5.1 Preliminaries.** In [23], Denef and Vercauteren systematically lift the equation of $C'$ over $k$ to equations over $R$ enabling them to get a handle on the dagger ring of the coordinate ring of $C'$. Here is a (very) abbreviated account of the process.

We first focus on the polynomial $\overline{h}(x)$. Let $\overline{c}$ be its leading coefficient and let $\overline{P}_0(x), \ldots, \overline{P}_r(x)$ its distinct monic irreducible factors so that

$$\overline{h}(x) = \overline{c} \prod_{i=0}^{r} \overline{P}_i(x)^{t_i} \tag{3.11}$$

for some positive integers $t_i$. Let $c \in R$ be a lift (preimage) of $\overline{c}$ and $P_0(x), \ldots, P_r(x) \in R[x]$ monic lifts of $\overline{P}_0(x), \ldots, \overline{P}_r(x) \in k[x] = R[x]/2R[x]$. Set

$$\overline{H}(x) := \prod_{i=0}^{r} \overline{P}_i(x)$$

$$H(x) := \prod_{i=0}^{r} P(x)$$

$$h(x) := c \prod_{i=0}^{r} P_i(x)^{t_i}.$$

We make the assumption that $\overline{H}(x)|\overline{f}(x)$. In [23, §§ 3.2] it is shown that this assumption may be made without loss of generality.

Next lift $\overline{f}(x)$ to a monic polynomial $f(x) \in R[x]$ in such a way that $H(x)|f(x)$. This can be done, for example, by lifting the ratio $Q_{\overline{f}}(x) := \overline{f}(x)/\overline{H}(x)$ to a monic polynomial $Q_f(x)$ and then setting

$$f(x) := H(x)Q_f(x).$$

The coordinate ring of $C'$ is

$$\overline{A} := \frac{k[x, y, \overline{H}(x)^{-1}]}{(y^2 + \overline{h}(x)y - \overline{f}(x))}$$

and the ring

$$A = \frac{R[x, y, H(x)^{-1}]}{y^2 + h(x)y - f(x)}$$

is a natural $R$–algebra which one checks is flat over $R$ and satisfies $A/2A = \overline{A}$.

One checks that the elements of $A^\dagger$ can be represented by series of the form

$$\sum_{i=-\infty}^{\infty} (U_i(x) + V_i(x)y)\, S(x)^i \tag{3.12}$$

with the degree of $U_i(x)$ and $V_i(x)$ smaller than the degree of $S(x)$. Here $S(x) = H(x)$ if $\deg H > 0$ and $S(x) = x$ if $H(x) = 1$. We refer to [23, §§ 3.2] for the constraints on the 2–adic orders of $U_i(x)$ and $V_i(x)$ imposed by the growth conditions for over convergence in the dagger ring $A^\dagger$.

**3.5.2 Frobenius on the dagger ring.** Recall that we have a lift $\Sigma\colon R \to R$ of the 2–power Frobenius $\sigma\colon k \to k$. The map $\Sigma$ can be extended to an endomorphism $\Sigma\colon A^\dagger \to A^\dagger$ in many ways. Denef and Vercauteren's method is as follows: (a) map $x$ to $x^2$, i.e. set $x^\Sigma = x^2$ and then (b) (b) define $y^\Sigma$ by using Newton interpolation. In greater detail, we know that $y^\Sigma$ is a solution of

$$(y^\Sigma)^2 + h(x)^\Sigma y^\Sigma - f(x)^\Sigma = 0 \qquad y^\Sigma \equiv y^2 \bmod 2$$

In a suitable completion of $A$ one could use (**Newt**) to solve for $y^\Sigma$ (since we know the action of $\Sigma$ on $R$ and $x$, whence on polynomials in $x$ with coefficients in $R$). Denef and Vercauteren show that the end result is in $A^\dagger$. Note that required recursive relation on a sequence $\{W_k\}$ converging 2-adically to $y^\Sigma$ via (**Newt**) is

$$W_{k+1} \equiv W_k - \frac{W_k^2 + h(x)^\Sigma W_k - f(x)^\Sigma}{2W_k + h(x)^\Sigma} \qquad \bmod 2^{k+1}$$

It is not *a priori* clear how the above recursive process gives $y^\Sigma$ in the form (3.12). We wish to deconstruct the somewhat cryptic process given [23, Algorithm 2] for getting a good approximate for $y^\Sigma$ (without getting into the issues of bounds for the number of steps involved). First note that the Weil conjectures give us bounds for the coefficients of the characteristic polynomial $\chi$ and this allows us to compute $y^\Sigma$ up to a suitable power of 2 (one has to actually refine matters a bit). The exact bounds are as in *loc.cit.* For simplicity we assume that $\deg H > 0$ so that $S = H$ in what follows. If $P, Q, R, S \in K[x, H(x)^{-1}]$ then in what follows we implicitly use the formula

$$(P + Qy)(R + Sy) = (PR + QS \cdot f) + (PS + Q(R - Sh))y$$

dictated by the relation for the plane curve $C^\circ$. Let $t_i$ be as in (3.11), and let $D = \max_i t_i$. Then $h | H^D$. Let

$$Q_H(x) := H(x)^D / h(x)$$

and let

$$\lambda_k := \frac{Q_H^2}{H^{2D}}(2W_k + h^\Sigma - h^2).$$

The algorithm used by Denef and Vercauteren for computing $W_k$ recursively in the form (3.12) is as follows. First suppose we write

$$W_k = \alpha_k + \beta_k y.$$

Then the numerator in the expression for $W_{k+1} - W_k$ given earlier is

$$W_k^2 - h(x)^\Sigma W_k + f(x)^\Sigma = T'_A + T'_B y$$

where

$$T'_A = (\alpha_k + h^\Sigma)\alpha_k - \beta_k^2 f - f^\Sigma, \qquad T'_B = (2\alpha_k - h\beta_k + h^\Sigma)\beta_k.$$

The `Lift_Frobenius_y` algorithm in [23, Alg. 2] then computes

$$W_{k+1} = W_k - (Q_H^2 / H^{2D})(T'_A + T'_B y)(\gamma_{k+1} + \delta_{k+1} y)$$

where $\theta_k := \gamma_k + \delta_k y$ follows the recursion formula

$$\theta_{k+1} = 2\theta_k - (1 + \lambda_k)\theta_k^2$$

with $\theta_1 = 1$ and $\lambda_1 = (Q_H^2 / H^{2D})(2(f - hy) - h^\Sigma - h^2)$. To orient the reader we point out that for a general $\lambda$ the recurrence relation $\tau_{n+1} = 2\tau_n - (1 + \lambda)\tau_n^2$ with

initial condition $\tau_1 = 1$ gives us $\tau_n = \sum_{i=0}^{\psi}(n)(-1)^i\lambda^i$ where $\psi(n) = 2^{n-1} - 1$, whence, in the formal power ring, $\tau_n$ converges rapidly to

$$1/(1+\lambda)$$

On the other hand [23, Formula (16)] tells us that

$$W_{k+1} = W_k - (Q_H^2/H^{2D})(T_A' + T_B')(1+\lambda_k)^{-1}.$$

**3.5.3 Basis for $H^1(C)$.** Let $\overline{A}$ be the coordinate ring of $C'$ and let $A^\dagger$ be the corresponding dagger algebra over $R$ in §§ 3.5.1. To ease typographical load we will not distinguish between a closed differential form in $\widetilde{\Omega}^1_{A^\dagger}$ and the MW cohomology class in $H^1(C')$ it represents (i.e. the given closed form modulo exact differential forms). Let

$$S_C := \{y\mathrm{d}x, xy\mathrm{d}x, \ldots, x^{2g-1}y\mathrm{d}x\} \subset H^1(C')$$

and

$$S_P := \left\{\frac{\mathrm{d}x}{H(x)}, \frac{x\mathrm{d}x}{H(x)}, \ldots, \frac{x^s\mathrm{d}x}{H(x)}\right\} \subset H^1(C')$$

In [23] it is shown that $S_C \cup S_P$ is a basis of $H^1(C')$. Moreover the above disjoint union reflects the decomposition (3.5). In particular

$$ydx, xydx, \ldots, x^{2g-1}ydx$$

gives a basis of $H^1(C)$ where we identify $H^1(C)$ with a subspace of $H^1(C')$ via the exact sequence (3.4). Moreover, the Frobenius map on $H^1(C)$ is obtained by restricting the Frobenius map on $H^1(C')$.

**3.5.4 Frobenius on the above basis.** We compute $F := F_1^C$ by first computing the action of the 2–power Frobenius—also denoted $\Sigma$—on the basis elements of $H^1(C)$ and then raising (with a twist which takes into account the fact that $\Sigma$ is not $K$–linear on $H^1(C)$) $\Sigma$ to the $a$–th power. Note that

$$(x^i y\mathrm{d}x)^\Sigma = x^{2i}y^\Sigma \mathrm{d}x^\Sigma = 2x^{2i+1}y^\Sigma \mathrm{d}x.$$

In order to write the right side as a linear combination of elements in $S_C$ we have to

- write $y^\Sigma$ in the form (3.12) as outlined in §§ 3.5.4, and then
- find a differential form which is in the span of $S_C$ which is equivalent modulo exact forms to the right side.

Assuming that the procedure given in `Lift_Frobenius_y` of [23] has been implemented (cf. §§ 3.5.4 above) the principal difficulty is reducing closed differential forms of the form

$$\sum T_i(x)S(x)^i y\mathrm{d}y \tag{3.13}$$

to a $K$–linear combination of elements in $S_C$. Here, the above sum is a priori an infinite Laurent series, but in practice, because of bounds implicit in the `Lift_Frobenius_y` algorithm, is a Laurent polynomial. We now describe briefly how the `Reduce_MW_Cohomology` algorithm in [23] addresses this issue. Again for simplicity we assume $\deg H > 0$ so that $S = H$.

For a polynomial $f(x)$ over $K$, the differential $f/H\mathrm{d}x$ is invariant under the hyperelliptic involution and hence its cohomology class lies in $P$. Therefore, looking at (3.13), we have to only worry about reducing $T_i(x)H(x)^i y\mathrm{d}x$ to required form. The problem is split into two cases—when $i \geq 0$ and when $i < 0$.

Suppose $i > 0$. The problem then is to reduce differentials of the form $\varphi(x)y\mathrm{d}x$ where $\varphi(x)$ is a polynomial over $K$. If $\deg\varphi < 2g$ we are done. So assume $\deg\varphi \geq 2g$. Suppose the leading coefficient of $\varphi$ is $\lambda$. One finds an exact differential

$$\omega = P(x)y\mathrm{d}x$$

where $P$ is a monic polynomial over $K$ such that $\deg P = \deg\varphi$. Then

$$\widetilde{\varphi}(x)y\mathrm{d}x := \varphi(x)y\mathrm{d}x - \lambda\omega$$

is cohomologous to $\varphi(x)y\mathrm{d}x$. Now note that $\deg\widetilde{\varphi} < \deg\varphi$. One iterates this process until we get a differential whose degree is less than $2g$. One therefore needs to find monic polynomials $P_{2g}(x), P_{2g+1}(x), \ldots$ such that for $k \geq 2g$, $\deg P_k = k$ and $P_k(x)y\mathrm{d}x$ is *exact*. The candidate offered by Denef and Vercauteren is

$$P_k = c(k)\left[x^{k-2g}(2f' + hh') + ((k-2g)/3)x^{k-2g-1}(4f + h^2)\right]$$

where

$$c(k) = (2(2g+1) + 4(k-2g)/3)^{-1}.$$

If $i < 0$ we rewrite $T_i(x)H(x)^i$ as $T_k(x)/H(x)^k$ where $k = -i > 0$ (we are aware of the resulting notational ambiguity, but it is a minor abuse of notation which ought to cause no confusion). Let $Q_h = h/H \in K[x]$. In [23] it is shown that

$$\frac{T_k(x)}{H(x)^k}y\mathrm{d}x \equiv \frac{A_k(x)}{H(x)^{k-1}}y\mathrm{d}x + \frac{B_k(kH'Q_h^2 - 6Q_f' - 3Q_hh') - B_k'(4Q_f + Q_hh')}{(6-4k)H^{k-1}}$$

where $A_k, B_k$ satisfy

$$T_k = A_k(x)H(x) + B_k(x)Q_f(x)H'(x).$$

The above decomposition is obtained by the Euclidean algorithm using the easily verifiable fact that $\gcd(Q_f, H) = 1$.

To obtain the matrix of $F_1^C : H^1(C) \to H^1(C)$ on the basis $S_C$ first compute the matrix $M(i,j)$ of the $\mathbb{Q}_p$–linear map $\Sigma$ on $H^1(C)$, and then get $F_1^C$ by computing

$$M_{\mathcal{F}} := MM^{\Sigma}\ldots M^{\Sigma^{a-1}}$$

One then computes the characteristic polynomial $\chi$ of $M_{\mathcal{F}}$ and uses (**Zeta-X**) to compute $Z(C, t)$.

## 3.6 Data structures

In this section we will discuss the data structures used to implement the algorithm in C++. We use the structures provided by NTL (Number Theory Library) whenever possible. However, we need to define our own structures in order to represent elements of $R$, $K$, $R[x]$ and $K[x]$.

**3.6.1 Representing elements of $R$ and $R[x]$.** Recall that $R = \frac{\mathbb{Z}_2[T]}{P(T)} = \mathbb{Z}_2[\Theta]$ where $P(T)$ is a monic lift to $\mathbb{Z}[T] \subset \mathbb{Z}_2[T]$ of a $q$-th degree monic irreducible polynomial $P$ over $\mathbb{F}_2$ (and $\Theta$ is the coset $T + (P(T)) \in \mathbb{Z}_2/(P(T))$. In practice, since we compute over $R/2^NR$ for a suitable integer $N$, we will identify $R$ with an extension of $\mathbb{Z}$ thus:

$$R = \frac{\mathbb{Z}[T]}{P(T)} = \mathbb{Z}[\Theta].$$

NTL provides the class ZZX, which represents elements of $\mathbb{Z}[x]$. So, we represent an element of $R$ as a ZZX with addition and subtraction as provided by ZZX but with multiplication performed modulo $P$.

Next, recall that

$$R[x] = \frac{\mathbb{Z}[T][x]}{P(T)} = \mathbb{Z}[\Theta][x]$$

with $P$ as before.

We begin by defining the class ZZXY, which represents elements of $\mathbb{Z}[x][y]$. Each such element is represented as a vector of ZZXs. Addition and subtraction are defined in the obvious way, and the classical ($O(n^2)$) algorithm is used for multiplication.

We then define the class RX as an extension of ZZXY. RX represents elements of $R[x]$. Addition, subtraction, and multiplication are defined as in ZZXY. However, whenever we need to multiply coefficients of an RX (e.g. when multiplying two RXs), we perform this multiplication modulo $P$.

**3.6.2 Representing elements of $K$ and $K[x]$.** Recall that $K = \frac{\mathbb{Q}_2[T]}{P(T)}\mathbb{Q}_2[\Theta]$ and as before, since our computations are modulo a power of 2 we may, in practice assume

$$K = \frac{\mathbb{Q}[T]}{P(T)} = \mathbb{Q}[\Theta]$$

We represent an element of $K$ as an element of $R$ along with an integer denominator. However, the denominator is always a power of 2, so it is sufficient to just store the integer $k$ such that the denominator is $2^k$.

Next, recall that (in practice)

$$K[x] = \mathbb{Q}[\Theta][x]$$

In order to represent elements of $K[x]$, we modify the class RX so that it also includes an integer denominator. Again, since the denominator is a power of 2, it is sufficient to store the integer $k$ such that the denominator is $2^k$. Note that for simplicity, we store a single denominator for each element of $K[x]$, rather than storing a denominator for each coefficient of each element of $K[x]$.

**3.6.3 Representing Laurent polynomials in $S \in R[x]$.** We define the class RXSeries to represent Laurent polynomials in a monic polynomial $S \in R[x]$. Each element is a represented as a vector of RXs where each such RX has degree less than $S$. In order to accommodate terms of negative degree, each RXSeries also includes a shift value. Addition and subtraction are defined as usual, and Karatsuba's algorithm is used for multiplication. After each multiplication, we must ensure that each coefficient in the resulting series has degree less than $S$. This is accomplished by dividing each coefficient by $S$, keeping only the remainder while adding the quotient to the coefficient of the next (higher degree) term.

## 3.7 Algorithm for lifting the curve to characteristic zero

**Algorithm** (`Lift_Curve`)

**IN:** *Hyperelliptic curve $\bar{C}$ over $F_q$ given by equation $y^2 + \bar{h}(x)y = \bar{f}(x)$.*
**OUT:** *Curve $C : y^2 + h(x)y = f(x)$ over $R$, polynomial $H(x) \in R[x]$ with $H|h$ and $H|f$, and $D \in N$ such that $h|H^D$.*

1. $\bar{c} =$ `LeadCoeff`$(\bar{h})$;
2. $\bar{h} = \bar{h}/\bar{c}$;
3. Let $\bar{g}_1, \bar{g}_2, ..., \bar{g}_n$ be the irreducible factors of $\bar{h}$ and let $b_1, b_2, ..., b_n$ be integers such that $\bar{h} = \bar{g}_1^{b_1} \cdot \bar{g}_2^{b_2} \cdot ... \cdot \bar{g}_n^{b_n}$;

4. $c = \texttt{Lift}(\bar{c})$;
5. For $i = 1$ to $n$ Do
   5.1 $g_i = \texttt{Lift}(\bar{g}_i)$;
6. $\bar{H} = \bar{g}_1 \cdot \bar{g}_2 \cdot ... \cdot \bar{g}_n$;
7. $H = g_1 \cdot g_2 \cdot ... \cdot g_n$;
8. $h = g_1^{b_1} \cdot g_2^{b_2} \cdot ... \cdot g_n^{b_n}$;
9. $D = \max(b_1, b_2, ..., b_n)$;
10. $Qf = \texttt{Lift}(\bar{f}/\bar{H})$;
11. $f = H \cdot Qf$;
12. Return $h$, $f$, $H$, $D$.

## 3.8 Inversion

**Algorithm** (Invert_R)

**IN:** $a \in \frac{\mathbb{Z}[\theta]}{P(\theta)}$ *such that* $2 \nmid a$, *and precision* $N$.
**OUT:** $b \in \frac{\mathbb{Z}[\theta]}{P(\theta)}$ *such that* $a \cdot b \equiv 1 \bmod 2^N$.

1. $M = \lceil \log_2 N \rceil$;
2. Let $b$ be such that $a \cdot b \equiv 1 \bmod 2$;
3. For $i = 1$ to $M$ Do
   3.1 $r \equiv a \cdot b \bmod 2^{2^i}$;
   3.2 $s = (1 - r)/(2^{2^{i-1}})$;
   3.3 $q = 2^{2^{i-1}} \cdot s + 1$;
   3.4 $b \equiv b \cdot q \bmod 2^{2^i}$;
4. Return $b \equiv b \bmod 2^N$.

The main idea of this algorithm is that at the start of each iteration of the For loop, we have $a \cdot b \equiv 1 \bmod 2^{2^{i-1}}$ (and it is obvious that this is true at the start of iteration 1). So we have $a \cdot b - 1 = k \cdot 2^{2^{i-1}}$ for some $k \in \mathbb{Z}$. Then, after step 3.1, we have $r = 1 + k \cdot 2^{2^{i-1}}$. After step 3.2, $s = -k$. So after step 3.3, we have $q = -k \cdot 2^{2^{i-1}} + 1$. At this point, we have $a \cdot b \cdot q = (k \cdot 2^{2^{i-1}} + 1)(-k \cdot 2^{2^{i-1}} + 1) = 1 - k \cdot 2^{2^i}$. That is, $a \cdot b \cdot q \equiv 1 \bmod 2^{2^i}$. Then, by letting $b = b \cdot q$ in step 3.4, we have $a \cdot b \equiv 1 \bmod 2^{2^i}$ at the end of each iteration of the For loop.

## 3.9 The 2–power Frobenius on $K$

**Algorithm** (Lift_Frobenius_$\Theta$)

**IN:** $\bar{P} \in \mathbb{F}_2[x]$ *such that* $P$ *is the minimal polynomial of* $\Theta$, *and precision* $N$.
**OUT:** $\Theta^{\Sigma}$.

1. $r = \Theta^2$;
2. $B = \lceil \log_2 N \rceil$;
3. For $i = 1$ to $B$ Do
   3.1 $r = r - P(r) \cdot \texttt{Invert_R}(P'(r), 2^i) \bmod 2^{2^i}$;
4. Return $r \bmod 2^N$.

This algorithm uses Newton interpolation, as discussed in Sections 3.2 and 3.3.

## 3.10 The characteristic polynomial of Frobenius

**Algorithm** (`Characteristic_Pol`)

**IN:** *$n \times n$ matrix $M$ whose entries are in $R$, and precision $N$.*
**OUT:** *Characteristic polynomial of $M$.*

1. $A = \lambda \cdot I - M$; Refer to the entries of $A$ as $a_{i,j}$, $1 \leq i, j \leq n$;
2. `For` $c = 1$ `to` $n$ `Do`
   2.1 Write each entry below the diagonal in column $c$ of $A$ as $2^k \cdot \alpha$, with $2 \nmid \alpha$, and say the smallest such $k$ is found in row $i$;
   2.2 Swap row $i$ and row $c+1$ of $A$;
   2.3 Swap column $i$ and column $c+1$ of $A$;
   2.4 Let $\alpha$ and $k$ be such that $a_{c+1,c} = 2^k \cdot \alpha$ with $2 \nmid \alpha$;
   2.5 `For` $r = c+2$ `to` $n$ `Do`
      2.5.1 Let $\beta$ and $j$ be such that $a_{r,c} = 2^j \cdot \beta$ with $2 \nmid \beta$;
      2.5.2 $t_r = 2^{j-k} \cdot$ `Invert_R`$(\alpha, N) \cdot \beta$;
      2.5.3 (row $r$ of $A$) = (row $r$ of $A$) $- t_r \cdot$ (row $c+1$ of $A$);
   2.6 `For` $q = c+2$ `to` $n$ `Do`
      2.6.1 (col $c+1$ of $A$) = (col $c+1$ of $A$) $+ t_q \cdot$ (col $q$ of $A$);
3. $v =$ (col $n$ of $A$); Refer to the entries of $v$ as $v_i$, $1 \leq i \leq n$;
4. `For` $c = n-1$ `down to` $1$ `Do`
   4.1 `For` $r = 1$ `to` $c$ `Do`
      4.1.1 $v_r = a_{r,c} \cdot v_{c+1} - a_{c+1,c} \cdot v_r$
5. `Return` $v_1$.

In step 2 of this algorithm, we put $A$ in upper Hessenberg form, with $\lambda$ appearing only on the diagonal. Upper Hessenberg form is similar to upper triangular form, except that the entries immediately below the diagonal may be non-zero. Step 2.5 clears all entries beneath row $c+1$ in column $c$ - this may introduce $\lambda$ in entries below the diagonal in column $c+1$. Then, step 2.6 clears $\lambda$ from entries below the diagonal in column $c+1$. Finally, in step 3, we compute the determinant of $A$. The overall complexity of this algorithm is $O(n^3)$ multiplications in $R$. By contrast, the complexity of the classical determinant-based algorithm is $O(n!)$ multiplications in $R$.

## 3.11 Multiplication

In this section, we will discuss the implementation of multiplication for $R[x]$ and for Laurent polynomials in $S \in R[x]$.

**3.11.1 Multiplying elements of $R[x]$.** Empirically, the overwhelming majority of the running time of the `Hyperelliptic_Zeta_Function` algorithm is spent in step 3 (that is, in `Lift_Frobenius_y`). In `Lift_Frobenius_y`, elements of $R[x]$ appear primarily as coefficients of Laurent Series. These coefficients have degree less than $g$. This means that most of the elements of $R[x]$ that we deal with have small degree. For polynomials of small degree, the classical $O(n^2)$ multiplication algorithm is the most efficient, and this is what we implement.

**3.11.2 Multiplying Laurent polynomials in $S \in R[x]$.** In the `Lift_Frobenius_y` algorithm, we need to multiply lengthy Laurent polynomials in

$S \in R[x]$. As these polynomials have hundreds of terms, the classical multiplication algorithm is very inefficient. We instead implement Karatsuba's $O(n^{\log_2 3})$ algorithm for multiplication.

Let $p(x)$ and $q(x)$ be polynomials of degree $2m-1$ for some $m$ a power of 2. We can write $p$ and $q$ as:

$$p = p_1 \cdot x^n + p_0$$
$$q = q_1 \cdot x^n + q_0$$

where $p_0(x)$, $p_1(x)$, $q_0(x)$, and $q_1(x)$ are polynomials of degree $m-1$. Then, note that:

$$\begin{aligned} p \cdot q &= (p_1 \cdot x^n + p_0) \cdot (q_1 \cdot x^n + q_0) \\ &= p_1 \cdot q_1 \cdot x^{2n} + (p_1 \cdot q_0 + p_0 \cdot q_1) \cdot x^n + p_0 \cdot q_0 \end{aligned}$$

That is, a single multiplication of degree $2m-1$ polynomials can be reduced to four multiplications of degree $m-1$ polynomials. So far, however, we have gained nothing since if we keep recursively applying this process to perform multiplication (until we are left with degree 0 polynomials that we multiply directly), we will have an algorithm whose asymptotic time complexity is identical to that of the classical algorithm. We can do better by noting that:

$$p \cdot q = p_1 \cdot q_1 \cdot x^{2n} + [(p_1 + p_0) \cdot (q_1 + q_0) - p_1 \cdot q_1 - p_0 \cdot q_0] \cdot x^n + p_0 \cdot q_0 \quad (3.14)$$

That is, a single multiplication of degree $2m-1$ polynomials can in fact be reduced to just *three* multiplications of degree $m-1$ polynomials. Karatsuba's algorithm involves recursively applying this process to perform multiplication.

When implementing Karatsuba's algorithm, we drop the requirement that the two polynomials have equal degree $2m-1$ for some $m$ a power of 2. If the polynomials being multiplied are not of equal degree, we simply pad the lower degree polynomial with leading zero coefficients so that the two polynomials have the same number of terms. If the polynomials being multiplied are of even degree $2m$ , we split each polynomial into one polynomial of degree $m$ and another of degree $m-1$ (rather than splitting into two polynomials of degree $m-1$).

Our implementation of Karatusba's algorithm for RXSeries (recall that this is the class that represents Laurent polynomials in $S \in R[x]$) is based on NTL's implementation of Karatsuba's algorithm for ZZX (this class represents $\mathbb{Z}[x]$). In particular, when multiplying RXSeries we preallocate the memory required for the intermediate results produced by recursive calls (this is much more efficient than allocating and deallocating memory within each recursive call) and we reuse this preallocated memory whenever possible (that is, when an intermediate result is no longer needed, we try to reuse the memory that it occupies).

## 3.12 Running times

Table 1 provides running times for our implementation. We tested our implementation on an AMD XP 1600+ processor. We include times for genus 2, 3, and 4 Artin-Schreier and non-Artin-Schreier curves with Jacobians with order of 120, 144, 168, 192 and 216 bits. When comparing our results to those in [23, §§ 5.1], note that Denef and Vercauteren test their implementation on an AMD XP 1700+ processor. More significantly, Denef and Vercauteren implement their algorithm in `C` and use assembly language for basic operations on integers modulo $2^N$ for $N \leq 256$, while our implementation is in `C++` and relies on NTL (Number Theory Library)

and GMP (GNU Multiprecision Library) for basic operations. Furthermore, Denef and Vercauteren use a combination of Karatsuba's and Toom's algorithms for multiplying Laurent polynomials over $R[X]$, while we use only Karatsuba's algorithm.

**Table 1** Running time (in seconds) for genus 2, 3, and 4 Artin-Schreier (AS) and non-Artin-Schreier (non-AS) hyperelliptic curves over $\mathbb{F}_{2^n}$

| Size of Jacobian | Genus 2 curves | | Genus 3 curves | | Genus 4 curves | |
|---|---|---|---|---|---|---|
| $gn$ | AS | non-AS | AS | non-AS | AS | non-AS |
| 120 | 72 | 156 | 106 | 239 | 114 | 247 |
| 144 | 171 | 309 | 159 | 373 | 211 | 462 |
| 168 | 259 | 455 | 271 | 536 | 325 | 705 |
| 192 | 378 | 629 | 404 | 699 | 486 | 988 |
| 216 | 492 | 989 | 684 | 1213 | 605 | 1281 |

## 3.13 Parallelization

In this section we will discuss our parallelized implementation of the algorithm. Our implementation uses the MPI (Message Passing Interface) library, and uses four processors. It is based on the single-processor implementation discussed elsewhere in this report, but with the `Lift_Frobenius_y` algorithm and a portion of the `Hyperelliptic_Zeta_Function` algorithm parallelized.

**3.13.1 Lift_Frobenius_y.** Some of the computations performed in step 4 (by far the most time-consuming step) of the `Lift_Frobenius_y` algorithm can be performed independently of each other. In particular, steps 4.1, 4.2, 4.3, and 4.4 can be performed simultaneously (one such step on each processor). Furthermore, steps 4.5 and 4.6 can be performed simultaneously, as can steps 4.7 and 4.8, as well as steps 4.9 and 4.10. Since we have four processors available, we split each of steps 4.5 through 4.10 into two halves which can be performed on separate processors. As a result, we make use of four processors throughout step 4.

As specified in [23], `Lift_Frobenius_y` returns two Laurent polynomials - $\alpha$ and $\beta$. However, $\alpha$ is not used by the `Hyperelliptic_Zeta_Function` algorithm, so our implementation of `Lift_Frobenius_y` does not return $\alpha$. This allows us to skip step 4.9 on the last iteration of the loop in step 4. We then split step 4.10 into four parts, one for each processor.

**3.13.2 Hyperelliptic_Zeta_Function.** In step 4 of `Hyperelliptic_Zeta_Function`, $2g$ calls are made to `Reduce_MW_Cohomology`. These calls can be performed independently of each other, and so we assign approximately one-fourth of these calls to each of our processors.

**3.13.3 Exchanging information.** The processors need to receive inputs needed to perform their assigned computations and the processors need to communicate their computed results. MPI provides built-in support for exchanging primitive type (such as long integers). It also provides support for exchanging strings (i.e., text). So, in order to exchange instances of RX and RXSeries, we convert them to strings using the extraction operator (`<<`), transmit these strings using MPI, and then (on the receiving end) convert the strings back to instances of RX

and RXSeries using the insertion operator (>>). This is not the most space-efficient way to transmit RX and RXSeries (since numbers are represented as ASCII characters rather than being represented in binary) but it is simple to implement and easy to debug. Using a binary representation would save space (and hence reduce transmission time), but this would be (at least partially) offset by the time required to produce the binary representation and then decipher it on the receiving end.

**3.13.4 Running times.** Table 2 provides running times for our implementation. We tested our implementation on a cluster with four AMD XP 1700+ processors. We include times for genus 2, 3, and 4 Artin-Schreier and non-Artin-Schreier curves with Jacobians of 120, 144, 168, 192, and 216 bits.

**Table 2** Running time (in seconds) of the parallelized implementation for genus 2, 3, and 4 Artin-Schreier (AS) and non-Artin-Schreier (non-AS) hyperelliptic curves over $\mathbb{F}_{2^n}$

| Size of Jacobian | Genus 2 curves | | Genus 3 curves | | Genus 4 curves | |
|---|---|---|---|---|---|---|
| $gn$ | AS | non-AS | AS | non-AS | AS | non-AS |
| 120 | 31 | 59 | 45 | 89 | 45 | 88 |
| 144 | 65 | 114 | 63 | 137 | 78 | 159 |
| 168 | 104 | 167 | 105 | 196 | 119 | 242 |
| 192 | 150 | 228 | 152 | 253 | 177 | 336 |
| 216 | 198 | 353 | 254 | 437 | 235 | 437 |

**Zubair Ashraf**
Security Services @ CRES Technologies
Markham, ON Canada L3S 4T9
zadamjee@yahoo.ca

**Ali Juma**
Department of Computer Science
University of Toronto
Toronto, ON Canada M5S 3G4
ajuma@cs.toronto.edu

**Pramathanath Sastry**
Chennai Mathematical Institute
Siruseri, TN-603102, India
pramath@cmi.ac.in

Fields Institute Communications
Volume **58**, 2010

CHAPTER 4

# An Introduction to Gröbner Bases

Mohammed Radi-Benjelloun

**Abstract.** Gröbner bases were introduced in 1965 by Bruno Buchberger. The basic idea behind the theory can be described as a generalization of the theory of polynomials in one variable. According to the Hilbert basis theorem, the polynomial ring $k[x]$, where $k$ is a field, is finitely generated. Given any set of generators $\{f_1, ....f_s\} \subseteq k[x]$ for $I$, one can compute (using the Euclidean Algorithm) a single polynomial $d = \gcd(f_1, ..., f_s)$ such that $I =< f_1, ..., f_s >=< d >$. Then a polynomial is in $I$ if and only if the remainder of the division of $f$ by $d$ is zero. Gröbner bases are the analog of greatest common divisors in the multivariate case in the following sense. A Gröbner basis of an ideal $I \subseteq k[x]$ generates $I$, and a polynomial $f \in k[x]$ is in $I$ if and only if the remainder of the division of $f$ by the polynomials in the Gröbner basis is zero. These kind of ideas had been around before Buchberger's work. But the true significance of Gröbner bases is the fact that they can be computed. Buchberger's Algorithm is what gave Gröbner basis theory the status of a subject in its own right. Gröbner basis theory is generating increasing interest because of its usefulness in providing tools which are applicable to a wide range of problems in mathematics, engineering, and computer science.

In this self-contained article we give an introduction to the basic theory of Gröbner bases. In the first chapter we introduce the kind of problems that motivate the use of Gröbner bases. Then we solve these problems in the linear and one variable cases by using the usual row reduction of matrices and the division algorithm respectively. In the second chapter we introduce monomial ideals and generalize the division algorithm to the n variables case. This leads us to define the central notion in this report namely, the notion of Gröebner bases. Buchberger's Algorithm which made Gröbner bases become a fundamental tool in computer algebra is then presented. Some examples illustrate the construction of Gröbner bases and the refinement of their definition.

## 4.1 Introduction

**4.1.1 Varieties and ideals.** Let $k$ be any field. We consider $k[x_1, ...x_n]$, the ring of polynomials in $n$ variables. For a positive integer $n$, we define the *affine*

*n-space* as

$$k^n = \{ (a_1, ..., a_n) \mid a_i \in k, i = 1, ..., n \}.$$

For $f \in k[x_1, ..., x_n]$, we define $V(f)$ to be the set of solutions of the equation $f = 0$, i.e.,

$$V(f) = \{ (a_1, ..., a_n) \in k^n \mid f(a_1, ..., a_n) = 0 \} \subseteq k^n.$$

$V(f)$ is called the variety defined by $f$. For example, the variety $V(x^2+y^2-1) \subseteq \mathbb{R}^2$ is the circle in the $xy$-plane with center $(0,0)$ and radius 1. More generally, given $f_1, ..., f_s \in k[x_1, ..., x_n]$, the variety

$$V(f_1, ..., f_n) = \{ (a_1, ..., a_n) \in k^n \mid f_i(a_1, ..., a_n) = 0, \ i = 1, 2..., s \}$$

is defined to be the set of all solutions of the system

$$f_1 = 0, f_2 = 0, ..., f_s = 0. \tag{4.1}$$

Note that $V(f_1, ..., f_s) = \bigcap_{i=1}^{s} V(f_i)$.

For example, the variety $V(x^2 + y^2 - 1, x - 3y^2) \subseteq \mathbb{R}^2$ is the intersection of the circle $x^2 + y^2 = 1$ and the parabola $x = 3y^2$ in the $xy$-plane. More generally, if $S \subseteq k[x_1, ..., x_n]$, we define

$$V(S) = \{ (a_1, ..., a_n) \in k^n \mid f(a_1, ..., a_n) = 0 \text{ for all } f \in S \}.$$

In general a variety can be the solution set of a system of equations such as (4.1) and the computation of the solutions can be improved drastically if the given system of equations is transformed into a different system that has the same solution but is "easier" to solve. This will be done by considering the ideal generated by the polynomials $f_1, ..., f_s$:

$$< f_1, .., f_s >= \left\{ \sum_{i=1}^{s} u_i f_i \mid u_i \in k[x_1, ..., x_n], \ i = 1, .., s \right\}.$$

The set $\{f_1, ..., f_s\}$ is called a *generating set* of the ideal $I =< f_1, ..., f_s >$. A "better" representation for the variety $V(f_1, ..., f_s)$ will be a better generating set for the ideal $< f_1, ..., f_s >$. In fact, one can easily check that $V(I) = V(f_1, ..., f_s)$. We note that an ideal may have different generating sets, with different numbers of elements. For example, in $k[x, y]$, $< x+y, x >=< x, y >=< x+xy, x^2, y^2, y+xy >$. Now, if we have $I =< f_1, ..., f_s >=< f_1', ..., f_t' >$, then $V(f_1, ..., f_s) = V(f_1', ..., f_t')$. This means that the system $f_1 = 0, ..., f_s = 0$, has the same solution as the system $f_1' = 0, ..., f_t' = 0$, and hence a variety is determined by an ideal, not by a particular set of equations.

Conversely, consider a collection $V$ of points of the affine space $k^n$, we define the set $I(V)$ of polynomials in $k[x_1, ..., x_n]$ by

$$I(V) = \{ f \in k[x_1, ..., x_n] \mid f(a_1, ..., a_n) = 0 \ for \ all \ (a_1, ..., a_n) \in V \}.$$

The set $I(V)$ is actually an ideal in $k[x_1, ..., x_n]$, and the Hilbert Basis Theorem states that every ideal in $k[x_1, ...x_n]$ is finitely generated. Thus $I(V)$ can be put in the form $< f_1, ..., f_s >$ for some $f_1, ..., f_s \in k[x_1, ...x_n]$. Another consequence of this theorem is that if $\Lambda$ is an infinite set and for all $\lambda \in \Lambda$ we have a polynomial $f_\lambda \in k[x_1, ..., x_n]$, then the solution of the infinite system $f_\lambda = 0, \lambda \in \Lambda$ is in fact, the solution set of a finite system, namely, of a finite generating set for the ideal $< f_\lambda, \lambda \in \Lambda >$. In order to find "better" generating sets for ideals, we will need to determine if two finite sets $\{ f_1, .., f_s \} \subseteq k[x_1, ..., x_n]$, and $\{ f_1', ..., f_t' \} \subseteq k[x_1, ..., x_n]$ generate the same ideal $I =< f_1, ..., f_s >=< f_1', ..., f_t' >$. This leads us to solve

the following problems: given $I =< f_1, ..., f_s >$ and $f \in k[x_1, ..., x_n]$, we get the following problems:

Problem 1. Determine whether $f \in I$. This is the so-called: "The ideal membership problem".
Problem 2. If $f \in I$, determine $u_1, ..., u_s \in k[x_1, ..., x_n]$ such that $f = u_1 f_1 + u_2 f_2 + \cdots + u_s f_s$.

**4.1.2 The linear case.** We consider the system

$$f_1 = 0,\ f_2 = 0,\ ..., f_s = 0, \text{ where each } f_i \text{ is linear.} \tag{4.2}$$

We use the row reduction method to change the system to row echelon form.

**Example 4.1** Let $f_1 = x + y - z$ and $f_2 = 2x + 3y + 2z$ be linear polynomials in $\mathbb{R}[x, y, z]$. We consider the ideal $I =< f_1, f_2 >$ and the variety $V(f_1, f_2)$, that is, the solutions to the system

$$\begin{cases} x + y - z = 0 \\ 2x + 3y + 2z = 0 \end{cases}. \tag{4.3}$$

We perform row reduction on the matrix associated with the system:

$$\begin{bmatrix} 1 & 1 & -1 \\ 2 & 3 & 2 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 4 \end{bmatrix}.$$

The last matrix is in row echelon form. The solutions of the system (4.3) are the same as those of the following system:

$$\begin{cases} x + y - z = 0 \\ \quad\ \ y + 4z = 0. \end{cases} \tag{4.4}$$

Now the solutions are easily obtained parametrically as : $x = 5z$ and $y = -4z$.

The row reduction process is in fact, a method to change a generating set for the ideal $I =< f_1, f_2 >$ into another generating set $< f_1, f_3 >$ with $f_3 = f_2 - 2f_1$. One can check that $I =< f_1, f_2 >=< f_1, f_3 >$. The process by which the polynomial $f_2$ was replaced by $f_3$ is called reduction of $f_2$ by $f_1$ and is denoted:

$$f_2 \xrightarrow{f_1} f_3.$$

The new polynomial $f_3$ that was created can be viewed as a remainder of a certain division: we used the first term of $f_1$, namely $x$ to eliminate a term from $f_2$, namely $2x$. Since the first term of $f_1$ cannot eliminate any other terms, the division stops and the remainder is exactly $f_3$. This can be written in long division form which gives rise to $f_2 = 2f_1 + f_3$. When the system has more than two equations, then division (or reduction) of a polynomial may require more than one polynomial.

**Example 4.2** Let $f_1 = y - z$, $f_2 = x + 2y + 3z$, and $f_3 = 3x - 4y + 2z$, be linear polynomials in $\mathbb{Q}[x, y, z]$. We consider the ideal $I =< f_1, f_2, f_3 >$ and the variety $V(f_1, f_2, f_3)$ that is the solutions to the system

$$\begin{cases} \quad\ \ y - z = 0 \\ x + 2y + 3z = 0 \\ 3x - 4y + 2z = 0. \end{cases} \tag{4.5}$$

The row reduction is as follows:

$$\begin{bmatrix} 0 & 1 & -1 \\ 1 & 2 & 3 \\ 3 & -4 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & -1 \\ 1 & 2 & 3 \\ 0 & -10 & -7 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & -1 \\ 1 & 2 & 3 \\ 0 & 0 & -17 \end{bmatrix}.$$

This says that a new generating set for $I =< f_1, f_2, f_3 >$ is $< f_1, f_2, -17z >$. Note that the polynomial $-17z$ is obtained by the following reductions:

$$f_3 \xrightarrow{f_2} -10y - 7z \xrightarrow{f_1} -17z.$$

This amounts to a division, similar to that in Example (4.1 ) of $f_3$ by $f_2$ and $f_1$ in succession which we denote $f_3 \xrightarrow{f_1,f_2} -17z$. Note that we have

$$f_3 = -10f_1 + 3f_2 - 17z. \tag{4.6}$$

We would like to "extract" from these examples some general ingredients that will be used in the general situation of non-linear polynomials.We will concentrate on Example 4.2. First, we imposed an order on the variables. We chose to eliminate $x$ first from the third equation (4.5) and then we chose to eliminate $y$ from the new third equation. That is, when we row reduce a matrix there is an order on how to proceed to introduce zeros: we first introduce zeros in the first column (eliminating $x$), and then introduce zeros in the second column (eliminating $y$), etc. We could have written the variables in a different order, say $f_1 = -z + y, f_2 = 3z + 2y + x$ and $f_3 = 2z - 4y + 3x$. We would have used the same row reduction method, but would have eliminated $z$ first, then $y$ and would have obtained a different set of equations in row echelon form, with the same solution set though. So the order does not matter, but there must be an order. We will see how this order is generalised (to the non-linear case). We note that in our example the order is such that $x$ is first followed by $y$ then $z$ and so the leading term of $f_1$ is $y$, the leading term of $f_2$ is $x$, and the leading term of $f_3$ is $3x$. Secondly, the reductions in Example 4.2. were obtained by subtracting multiples of $f_1$ and $f_2$. This had the effect of using the leading terms of $f_1$ and $f_2$ to eliminate terms in $f_3$ and in $-10y - 7z$ leaving the remainder $-17z$ and giving us equation (4.6). Note that -17z cannot be reduced any further (by using the leading term of $f_1$ and $f_2$).

**4.1.3 The one variable case.** In this section we consider polynomials in $k[x]$ and will use the Euclidean Algorithm to solve the problems mentioned in Section 4.1.1 The theory of polynomials in one variable is a good illustration of the more general theory that will be presented in the remainder of this report. Given a non-zero polynomial $f = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$, where $a_i \in k$, and $a_m \neq 0$, we recall that the degree of $f$, denoted $\deg(f)$ is $m$. The leading term of $f$, denoted $\mathrm{LT}(f)$, is $a_m x^m$. The leading coefficient of $f$, denoted $\mathrm{LC}(f)$, is $a_m$. Note that:

$$\deg(f) \leq \deg(g) \iff \mathrm{LT}(f) \textit{ divides } \mathrm{LT}(g).$$

**Proposition 4.3** *(The Division algorithm) Let $g$ be non-zero polynomial in $k[x]$. Then for any polynomial $f \in K[x]$, there exist $q$ and $r \in k[x]$ such that*

$$f = qg + r, \text{ with } r = 0 \text{ or } \deg(r) \lneq \deg(g).$$

*Moreover $r$ and $q$ are unique ($q$ is called the quotient and $r$ the remainder).*

The Algorithm follows.

Input : $f, g$

Output : $q, r$

$q := 0; r := f$

WHILE $r \neq 0$ AND LT($g$) divides LT($r$) DO

$$q := q + \frac{\mathrm{LT}(r)}{\mathrm{LT}(g)}$$

$$r := r - \frac{\mathrm{LT}(r)}{\mathrm{LT}(g)} g$$

**Definition 4.4** A Greatest Common Divisor of polynomials $f, g \in k[x]$ is a polynomial $h$ such that

(i) $h$ divides $f$ and $g$.

(ii) if $p$ is another polynomial which divides $f$ and $g$, then $p$ divides $h$.

We denote it $h = \gcd(f, g)$.

**Proposition 4.5** *(The Euclidean algorithm) Let $f, g \in k[x]$; then*

(i) $\gcd(f, g)$ *exists and is unique up to multiplication by a non-zero constant in $k$.*

(ii) $\gcd(f, g)$ *is a generator of the ideal* $< f, g >$ .

(iii) *There is an algorithm for finding* $\gcd(f, g)$*:*

Input : $f, g$

Output : $h = \gcd(f, g)$

$h := f$

$s := g$

WHILE $s \neq 0$ DO

  $rem :=$ remainder$(h, s)$

  $h := s$

  $s := rem$

For ideals expressed by more than two polynomials we have:

**Proposition 4.6** *Let $f_1, \ldots, f_s$ be polynomials in $k[x]$; then*

(i) $< f_1, \ldots, f_s > \; = \; < \gcd(f_1, \ldots, f_s) >$ .

(ii) *if $s \geq 3$; then* $\gcd(f_1, \ldots, f_s) \; = \; \gcd(f_1, \gcd(f_2, \ldots, f_s))$.

**Corollary 4.7** *If $k$ is a field, then every ideal of $k[x]$ is generated by one element. i.e. $k[x]$ is a PID (Principal Ideal domain).*

We can now solve the problems raised in Section (4.1.1) for the special case of polynomials in one variable.
To decide whether a polynomial $f$ is in the ideal $I =< f_1, \ldots, f_s >$ (or not), we first compute $g = \gcd(f_1, \ldots, f_s)$ by induction as in part (ii) of Proposition 4.6 , we then use the Division algorithm to divide $f$ by $g$. The remainder (of that division) is zero , if and only if $f$ is in the ideal $I =< f_1, \ldots, f_s > = < g >$.

## 4.2 Gröbner bases

**4.2.1 Monomial ideals.** In this chapter we consider a particularly simple type of ideals, called monomial ideals. A monomial ideal is an ideal generated by a set of monomials, i.e. elements in $k[x_1, ..., x_n]$ of the form $x_1^{e_1} x_2^{e_2} ... x_n^{e_n}$. Monomials are easier to understand and compute with than arbitrary polynomials.

For example, if

$$\underline{x}^a := x_1^{a_1} .... x_n^{a_n} \quad \text{and} \quad \underline{x}^b := x_1^{b_1} .... x_n^{b_n}$$

are two monomials then their greatest common divisor and least common multiple are:

$$\gcd(\underline{x}^a, \underline{x}^b) = x_1^{\min(a_1,b_1)} x_2^{\min(a_2,b_2)} ... x_n^{\min(a_n,b_n)}$$
$$\text{LCM}(\underline{x}^a, \underline{x}^b) = x_1^{\max(a_1,b_1)} x_2^{\max(a_2,b_2)} ... x_n^{\max(a_n,b_n)}.$$

and so it's trivial to see if a monomial divides another, and hence be in the same ideal for:

**Lemma 4.8** *Let $I$ be a monomial ideal, then*

$$\underline{x}^a := x_1^{a_1} .... x_n^{a_n} \in I \iff \exists\ \underline{x}^b := x_1^{b_1} .... x_n^{b_n} \in I \mid \underline{x}^b \text{ divides } \underline{x}^a$$

Moreover, Dickson's lemma states that all monomial ideals are finitely generated.

**Theorem 4.9** *(Dickson's lemma)*[21]. *Let $S$ be a set of monomials in $k[x_1, x_2, ..., x_n]$, and let $I =< S >$ be the monomial ideal generated by $S$, then there is a finite subset $S'$ of $S$ such that $< S > = < S' >$.*

**4.2.2 Monomial orderings.** As we have seen in the last two sections, it is important to specify an order on the monomials. In the linear case, we computed with $x$ first, then $y$, etc... In the one variable case, we used the highest degree term first(the leading term) in the division algorithm, so the order was: $x^{n+1} \succ x^n \succ \cdots \succ x^2 \succ x \succ 1$. In more than one variable we need an order analogous to the ones used in the two previous cases.

First let $\mathcal{M} := \{ x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} \mid a_i \in \mathbb{N},\ i = 1, ..., n \}$ denote the set of all monomials in $K[x_1, ..., x_n]$. We will introduce an order relation $\prec$ on $\mathcal{M}$. We define $\prec$ to be an *admissible order* or *monomial order* on $\mathcal{M}$ if it is a *total order* which is *compatible* with the multiplication of monomials, i.e.,

- For any pair of monomials we have $m \prec n$ or $n \prec m$ or $m = n$.
- If $m_1 \prec m_2$ and $m_2 \prec m_3$ then $m_1 \prec m_3$.
- $1 \prec m$ for any monomial $m \neq 1$.
- If $m_1 \prec m_2$ then $nm_1 \prec nm_2$ for any monomial $n \in \mathcal{M}$.

We next give some examples of admissible orderings. Note that, in small number of variables, we will use $x, y$ or $z$ instead of $x_1, x_2$ or $x_3$.

- the lexicographical ordering, Lex:
  $x_1^{i_1} \cdots x_n^{i_n} \prec x_1^{j_1} \cdots x_n^{j_n}$ If $i_1 = j_1, ..., i_k = j_k, i_{k+1} \lneq j_{k+1}$ for some $k$.
  In other words $m \prec n$ if the first variable with different exponents in $m$ and $n$ has lower exponent in $m$.
  For example, in $k[x, y, z]$, $x^2 z (= x^2 y^0 z) \prec x^2 y$; $y^{15} z^{30} \prec x$.
- The degree lexicographical ordering, Deglex (Also called: the graded lex order, grlex):
  $m = x_1^{i_1} \cdots x_n^{i_n} \prec x_1^{j_1} \cdots x_n^{j_n} = n$ If $\deg[m] = i_1 + \cdots + i_n \lneq j_1 + \cdots + j_n =$

deg$[n]$ or, if deg$[m]$ = deg$[n]$ and $m \prec n$ in Lex.
For example: $xyz \prec z^4$, and $x^2z^3 \prec x^2yz^2$.

- The degree reverse lexicographical ordering, Degrevlex (Also called: the graded reverse lex ord, grevlex):
  $m = x_1^{i_1} \cdots x_n^{i_n} \prec x_1^{j_1} \cdots x_n^{j_n} = n$ If $\deg[m] \lneq \deg[n]$, or if $\deg[m] = \deg[n]$ and $i_n = j_n$, $i_{n-1} = j_{n-1}, ..., i_k = j_k$, $i_{k-1} \gneq j_{k-1}$ for some $k$.
  In other words $m \prec n$ if $\deg[m] \lneq \deg[n]$ or if $\deg[m] = \deg[n]$ and the last variable with different exponents in $m$ and $n$ has higher exponent in $m$. For example $xyz^3 \prec x^2yz^2$.

There are different ways to vary these orderings to get new ones. All the examples we have given (so far) imply the ordering $x_n \prec x_{n-1} \prec \cdots \prec x_1$ of the variables. The ordering of the variables can be varied in $n!$ ways which all induce new orderings.

As a consequence of Dickson's Lemma, all classified orderings are well orderings in the sense of the following lemma.

**Lemma 4.10** *(Well orderings) Let $\prec$ be an admissible ordering on $\mathcal{M}$, then any nonempty subset $S$ of $\mathcal{M}$, has a smallest element: there is $m_0 \in S$ such that $m_0 \prec n$ for any $n \in S$.*

**4.2.3 Division algorithm in $k[x_1, ..., x_n]$.** In this section we define a division algorithm in $k[x_1, ..., x_n]$ that extends the algorithms seen in Section 4.1.2 and Section 4.1.3 .

The basic idea behind the algorithm is the same as we have seen for linear and one variable polynomials: when dividing $f$ by $f_1, \ldots, f_s$, we want to cancel terms of $f$ using the leading terms of the $f_i$'s and continue this process until it cannot be done anymore.

**Definition 4.11** Let $f = \sum_\alpha a_\alpha x^\alpha$ be a non-zero polynomial in $k[x_1, \ldots, x_n]$, Where $\alpha := (\alpha_1, ..., \alpha_n) \in \mathbb{Z}^n_{\geq 0}$ is a multi-index, and $\underline{x}^\alpha := x_1^{\alpha_1}, ..., x_n^{\alpha_n}$ is a monomial, and let $\succ$ be an admissible order on $\mathcal{M}$.

(i) The *multidegree* of $f$ is

$$\text{multideg}(f) = \max(\alpha \in \mathbb{Z}^n_{\geq 0} \ : \ a_\alpha \neq 0)$$

(the maximum is taken with respect to $\succ$).

(ii) The *leading coefficient* of $f$ is

$$\text{LC}(f) = a_{\text{multideg}(f)} \in k.$$

(iii) The *leading monomial* of $f$ is

$$\text{LM}(f) = x^{\text{multideg}(f)}.$$

(iv) The *leading term* of $f$ is

$$\text{LT}(f) = \text{LC}(f).\,\text{LM}(f).$$

**Example 4.12** Let $f = 4xy^2z + 4z^2 - 5x^3 + 7x^2z^2$ and let $\succ$ denote the Lex order. Then: multideg$(f) = (3,0,0)$, LC$(f) = -5$, LM$(f) = x^3$, LT$(f) = -5x^3$.

**Theorem 4.13** *Division Algorithm in $k[x_1, \ldots, x_n]$ Fix a monomial order on $\mathcal{M}$, and let $F = (\,f_1, \ldots, f_s\,)$ be an ordered s-tuple of polynomials in $k[x_1, \ldots, x_n]$. Then every $f \in k[x_1, \ldots, x_n]$ can be written as:*

$$f = a_1 f_1 + \cdots + a_s f_s + r,$$

*where* $a_i,\ r \in k[x_1, \ldots, x_n]$, *and either* $r = 0$ *or* $r$ *is a linear combination (with coefficients in* $k$*), of monomials none of which is divisible by any of* $\mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s)$. $r$ *is called a remainder of* $f$ *on division by* $F$. *Furthermore, if* $a_i f_i \neq 0$, *then we have* $\mathrm{multideg}(f) \geq \mathrm{multideg}(a_i f_i)$.

The algorithm follows:

Input : $f_1, \ldots, f_s, f$

Output : $a_1, \ldots, a_s,\ r$

$a_1 := 0; \ldots; a_s := 0; r := 0$

$p := f$

*WHILE* $p \neq 0$ *DO*

  $i := 1$

  divisionoccured := false

  WHILE $i \leq s$ AND divisionoccured = false DO

    IF $\mathrm{LT}(f_i)$ divides $\mathrm{LT}(p)$ THEN

$$a_i := a_i + \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)}.f_i$$

$$p := p - \frac{\mathrm{LT}(p)}{\mathrm{LT}(f_i)}$$

    *divisionoccured := true*

    ELSE $i := i + 1$

  IF divisionoccured = false THEN

    $r := r + \mathrm{LT}(p)$

    $p := p - \mathrm{LT}(p)$

**Example 4.14** Using the Lex order with $x > y$,we want to divide $f = x^2y + xy^2 + y^2$ by $f_1 = xy - 1$ and $f_2 = y^2 - 1$. Listing $f_1, f_2$ and the quotients $a_1, a_2$ vertically, we have the following setup:

$$\begin{array}{r|ccccccc}
a_1: & x & + & y & & \\
a_2: & & 1 & & & \\
xy-1 & x^2y & + & xy^2 & + & y^2 \\
y^2-1 & x^2y & - & x & & \\
\text{p=} & xy^2 & + & x & + & y^2 \\
 & xy^2 & - & y & & \\
 & x & + & y^2 & + & y \\
 & & & y^2 & - & 1 \\
 & & & y & + & 1
\end{array}$$

For $p\ =\ xy^2 + x + y^2,\ \ \mathrm{LT}(p) = y.\mathrm{LT}(f_1)$.

When $p = x + y^2 + y,\ \mathrm{LT}(p) = x$ is not divisible by any of $\mathrm{LT}(f_1)$ or $\mathrm{LT}(f_2)$, so we add it to the remainder $r := 0 + x$, and $p = y^2 + y$. Now $\mathrm{LT}(p) = y^2$ is divisible by $\mathrm{LT}(f_2)$ only, yielding $a_2 = 1$, and a new $p = y + 1$ which adds to $r := x + y + 1$; so we obtain: $x^2y + xy^2 + y^2 = (x + y)(xy - 1) + 1.(y^2 - 1) + x + y + 1$

**Example 4.15** Let's change the order of the divisors in the previous example. So $f = x^2y + xy^2 + y^2$ and $F = (f_1, f_2) = (y^2 - 1, xy - 1)$, we will keep the same

Lex order however . Applying the Algorithm above, we get:

$$x^2y + xy^2 + y^2 = (x+1)(y^2-1) + x.(xy-1) + 2x + 1.$$

Note that we have obtained a different remainder this time.

For $p = xy^2 + x + y^2$, $\mathrm{LT}(p) = y.\mathrm{LT}(f_1)$.

When $p = x + y^2 + y$, $\mathrm{LT}(p) = x$ is not divisible by any of $\mathrm{LT}(f_1)$ or $\mathrm{LT}(f_2)$, so we add it to the remainder $r := 0 + x$, and $p = y^2 + y$. Now $\mathrm{LT}(p) = y^2$ is divisible by $\mathrm{LT}(f_2)$ only, yielding $a_2 = 1$, and a new $p = y + 1$ which adds to $r := x + y + 1$; so we obtain:

$$x^2y + xy^2 + y^2 = (x+y)(xy-1) + 1.(y^2-1) + x + y + 1.$$

**Example 4.16** Let's change the order of the divisors in the previous example. The reason is that in the algorithm we choose $i$ to be the least $(i \leq s)$ such that $\mathrm{LT}(f_i)$ divides $\mathrm{LT}(p)$, but if we rearrange the $f_i$'s differently, we may get different $a_i$'s and a different remainder $r$. This raises another problem.

The ideal membership problem : $f \in < f_1, \dots, f_s >$ versus the remainder on division of $f$ by $F = (f_1, \dots, f_s)$ is zero. If $r = 0$, then we can write $f = a_1f_1 + \cdots + a_sf_s$ thus $f \in < f_1, \dots, f_s >$, but it is not necessary to have $r = 0$.

**Example 4.17** Let's take again the same family $f_1 = xy - 1, f_2 = y^2 - 1$, and let $f = xy^2 - x$. We use the Lex order as usual.

For $p = xy^2 + x + y^2$, $\mathrm{LT}(p) = y.\mathrm{LT}(f_1)$.

When $p = x + y^2 + y$, $\mathrm{LT}(p) = x$ is not divisible by any of $\mathrm{LT}(f_1)$ or $\mathrm{LT}(f_2)$, so we add it to the remainder $r := 0 + x$, and $p = y^2 + y$. Now $\mathrm{LT}(p) = y^2$ is divisible by $\mathrm{LT}(f_2)$ only, yielding $a_2 = 1$, and a new $p = y + 1$ which adds to $r := x + y + 1$; so we obtain:

$$x^2y + xy^2 + y^2 = (x+y)(xy-1) + 1.(y^2-1) + x + y + 1.$$

**Example 4.18** Let's change the order of the divisors in the previous example. If we divide $f$ by $(f_2, f_1)$, we get $f = x.f_2 + 0.f_1 + 0$ so that $f \in < f_1, f_2 >$. But we get $f = y.f_1 + 0.f_2 + (-x+y)$ if we divide it by $(f_1, f_2)$, obtaining thus a remainder $r \neq 0$ .

Buchberger solved this paradox in 1965 by finding a "better" generating set for the ideal $I = < f_1, \dots, f_s >$. For this set, the division does have a unique remainder, and $r = 0$ is necessary and sufficient for membership in the ideal $I$. He named this set after his advisor: Gröbner basis.

### 4.2.4 Gröbner bases.

**Definition 4.19** Let $I \subset k[x_1, \dots, x_n]$ be an ideal other than $\{0\}$.

(i) We denote by $\mathrm{LT}(I)$: the set of leading terms of elements of $I$. Thus

$$\mathrm{LT}(I) = \{\, cx^\alpha \mid \text{there exists } f \in I \text{ with } \mathrm{LT}(f) = cx^\alpha \,\}.$$

(ii) We denote by $< \mathrm{LT}(I) >$ the ideal generated by elements of $\mathrm{LT}(I)$.

**Remark 4.20** If we are given a finite set of generators of $I$, say $I = < f_1, \dots, f_s >$, then $< \mathrm{LT}(f_1), \dots, \mathrm{LT}(f_s) > \subset < \mathrm{LT}(I) >$ since $\mathrm{LT}(f_i) \in \mathrm{LT}(I)$, $i = 1, \dots, s$. But $< \mathrm{LT}(I) >$ can be strictly larger than $< \mathrm{LT}(f_1), \dots, \mathrm{LT}(f_s) >$.

**Example 4.21** Let $I =< f_1, f_2 >$, where $f_1 = x^3 - 2xy, f_2 = x^2y - 2y^2 + x$, and let $f = x.f_2 - y.f_1 = x^2$, so $\mathrm{LT}(f) = x^2 \in< \mathrm{LT}(I) >$ since $f \in I$. But $\mathrm{LT}(f) = x^2 \notin< x^3, x^2y >=< \mathrm{LT}(f_1), \mathrm{LT}(f_2) >$. Note that we are using the Deglex (grlex) order in this example.

**Proposition 4.22** *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal.*

(i) *$< \mathrm{LT}(I) >$ is a monomial ideal.*
(ii) *There are $g_1, \ldots, g_t \in I$ such that $< \mathrm{LT}(I) >=< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) >$.*

**Proof** (i) $< \mathrm{LT}(I) > \overset{def}{:=} < \mathrm{LT}(f), \text{for all}\, f \in I \setminus 0 >=< \mathrm{LM}(f), \text{for all}\, f \in I \setminus 0 >$ Since $\mathrm{LT}(f) = c.\mathrm{LM}(f)$ for all $f \in I$.

(ii) $< \mathrm{LT}(I) >$ is a monomial ideal(the above assertion), therefore, it is finitely generated by Dickson's Lemma (Theorem 4.9). That is, $< \mathrm{LT}(I) >=< \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_t) >$, for some $g_1, \ldots, g_t \in I$ which permits the conclusion, since (again) $\mathrm{LT}(g) = c.\mathrm{LM}(g)$ for all $g \in I$.

□

**Definition 4.23** A set of non-zero polynomials $G = \{g_1, \ldots, g_t\}$ contained in an ideal $I$, is called a Gröbner Basis for $I$, if and only if, for all $f \in I$ such that $f \neq 0$, there exists $i \in \{1, \ldots, t\}$ such that $\mathrm{LT}(g_i)$ divides $\mathrm{LT}(f)$. (i.e., $< \mathrm{LT}(I) >=< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) >$ . )

**Corollary 4.24** *Every non zero ideal $I \subset k[x_1, \ldots, x_n]$ has a Gröbner basis. (See Proposition 4.22(ii)).*

**Corollary 4.25** *If $G = \{g_1, \ldots, g_t\}$ is a Gröbner basis for the ideal $I$; then $I =< g_1, \ldots, g_t >$.*

Note that $< g_1, \ldots, g_t >\subset I$, since $G = \{g_1, \ldots, g_t\} \subset I$. Let $f \in I$, we can write $f = a_1g_1 + \cdots + a_tg_t + r$ where no term of $r$ is divisible by any of $\mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t)$, or $r = 0$ (by the Division Algorithm Theorem 4.13.). Thus $r = f - a_1g_1 - \cdots - a_tg_t \in I$, and $\mathrm{LT}(r) \in < \mathrm{LT}(I) >=< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) >$, hence $\mathrm{LT}(r)$ must be divisible by one of the $\mathrm{LT}(g_i)$, Contradicting the hypothesis on $r$, so $r = 0$, and $f \in< g_1, \ldots g_t >$.

**Remark 4.26** We get as a consequence of Corollary 4.24 and Corollary 4.25 that, every ideal $I \subset k[x_1, \ldots, x_n]$ is finitely generated. This paraphrases the classical and important result by the German mathematician David Hilbert in 1891:

**Theorem 4.27** *(Hilbert's Basis Theorem). Every ideal in a polynomial ring over a field is finitely generated.*

### 4.2.5 Properties of Gröbner bases.

**Proposition 4.28** *Let $G = \{g_1, \ldots, g_t\}$ be a Gröbner basis for an ideal $I \subset k[x_1, \ldots, x_n]$ and let $f \in k[x_1, \ldots, x_n]$. Then there is a unique $r \in k[x_1, \ldots, x_n]$ with the two following properties:*

(i) *No term of $r$ is divisible by any of $\mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t)$.*
(ii) *There is $g \in I$ such that $f = g + r$.*

*In particular, $r$ is the remainder on division of $f$ by $G$ no matter how the elements (of $G$) are listed when using the division Algorithm.*

**Proof** The division algorithm of $f$ by $G$ gives $f = a_1g_1 + \cdots + a_tg_t + r$, where $r$ satisfies (i). (ii) is also satisfied if we pose: $g = a_1g_1 + \cdots + a_tg_t \in I$. Let's prove that $r$ is unique. Suppose we have $f = g + r = g' + r'$ satisfy (i) and (ii), then $r - r' = g - g' \in I$. So if $r \neq r'$ , then $\mathrm{LT}(r - r') = \mathrm{LT}(g - g') \in < \mathrm{LT}(I) >$, and so it is divisible by one of the $\mathrm{LT}(g_i)$ by Theorem 4.8. This cannot happen, since no term of $r$ or $r'$ is divisible by one of the $\mathrm{LT}(g_i)$, Thus $r - r' = 0$ . □

**Corollary 4.29** *Let $G = \{g_1, \ldots, g_t\}$ be a Gröbner basis for an ideal $I \in k[x_1, \ldots, x_n]$ and let $f \in k[x_1, \ldots, x_n]$. Then $f \in I$ if and only if the remainder on division of $f$ by $G$ is zero.*

**Proof** given $f \in I$, write $f = f + 0$, so it satisfies the above proposition, and 0 is the remainder on division of $f$ by $G$.The converse is obvious. We can already answer the ideal membership problem by means of this corollary, if we know a Gröbner basis for the ideal $I$, or know how to find one. □

**Theorem 4.30** *Let $G = \{ g_1, \ldots, g_t \}$ be a set of non-zero polynomials in $k[x_1, \ldots, x_n]$. Then $G$ is a Gröbner basis if and only if for all $f \in k[x_1, \ldots, x_n]$, the remainder on the division of $f$ by $G$ is unique (independently of the order of division).*

**4.2.6 Construction of Gröbner bases.** We know that Gröbner bases exist, but this does not help much if we want to calculate them. We need a Criterion, to tell if a generating set for an ideal is a Gröbener basis. Once this criterion is given , an algorithm for determining them is constructed. We can already suggest that the remainder sometimes called "normal form" will be at the center of this criterion.

**4.2.7 $S$-Polynomials.** Let $I =< f_1, \ldots, f_s >$ be an ideal in $k[x_1, \ldots, x_n]$. $\{ f_1, \ldots f_s \}$ is not a Gröbner basis for $I$ if and only if

$$< \mathrm{LT}(I) > \neq < \mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s) >$$

(see Definition), i.e., there exists $f = h_1f_1 + \cdots + h_sf_s \in I$, such that $\mathrm{LT}(f) \notin < \mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s) >$, but this happens when the leading terms $\mathrm{LT}(h_if_i)$ cancel so that the leading term of $f$ is smaller. That's what happened in the Example 4.21. The simplest way in which this kind of cancellations occur is to use *S-polynomials.*

**Definition 4.31** Let $0 \neq f, g \in k[x_1, \ldots, x_n]$. Let $L = \mathrm{LCM}(\mathrm{LM}(f), \mathrm{LM}(g))$, for a given order $\prec$ . The polynomial:

$$S(f, g) = \frac{L}{\mathrm{LT}(f)}f - \frac{L}{\mathrm{LT}(g)}g$$

is called the *S-polynomial of $f$ and $g$.*

**Example 4.32** if $(f, g) = (x^2, xy - y^2)$ and the ordering is Lex, then $L = x^2y$, and $S(f, g) = \frac{x^2y}{x^2}f - \frac{x^2y}{xy}g = yf - xg = xy^2 = yx^2 - x(xy - y^2) = xy^2$. We see that $\mathrm{LT}(yf) = x^2y = \mathrm{LT}(xg)$ has been canceled in $S(f, g)$.

The last example shows that polynomial combination of $f$ and $g$ into *S-polynomial* produces cancellation of leading terms.
Conversely: Every cancellation of leading terms among polynomials of the same multidegree is produced by *S-polynomials* :

**Lemma 4.33** *Let* $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$ *be such that* $\mathrm{LM}(f_i) = X \neq 0$, *(all with the same leading monomial i.e. same multidegree).*

*Let* $f = \sum_{i=1}^{s} c_i f_i$ *with* $c_i \in k, i = 1, \ldots, s$ *; if* $\mathrm{LM}(f) \lneq X$ *(*$\mathrm{multideg}(f) \lneq \mathrm{multideg}(X)$*), then* $f$ *is a linear combination(with coefficients in* $k$*) of* $S(f_i, f_j)$, $1 \leq i \lneq j \leq s$.

**Proof** Write $f_i = a_i X +$ lower terms, $a_i \in k$ . Then the hypothesis says that $\sum_{i=1}^{s} c_i a_i = 0$, since the $a_i$'s are in $k$.

Now, by definition $S(f_i, f_j) = \frac{1}{a_i} f_i - \frac{1}{a_j} f_j$, since $\mathrm{LM}(f_i) = \mathrm{LM}(f_j) = X$. So we can write:

$$\begin{aligned} f &= c_1 f_1 + \cdots + c_s f_s \\ &= c_1 a_1 (\frac{1}{a_1} f_1) + \cdots + c_s a_s (\frac{1}{a_s} f_s) \\ &= c_1 a_1 (\frac{1}{a_1} f_1 - \frac{1}{a_2} f_2) + (c_1 a_1 + c_2 a_2)(\frac{1}{a_2} f_2 - \frac{1}{a_3} f_3) + \ldots \\ &\quad + (c_1 a_1 + \cdots + c_{s-1} a_{s-1})(\frac{1}{a_{s-1}} f_{s-1} - \frac{1}{a_s} f_s) + (c_1 a_1 + \cdots + c_s a_s) \frac{1}{a_s} f_s \\ &= c_1 a_1 S(f_1, f_2) + (c_1 a_1 + c_2 a_2) S(f_2, f_3) + \ldots \\ &\quad (c_1 a_1 + \cdots + c_{s-1} a_{s-1}) S(f_{s-1}, f_s), \end{aligned}$$

since $c_1 a_1 + \cdots + c_s a_s = 0$. □

**Theorem 4.34** *(Buchberger) Let* $G = \{ g_1, \ldots, g_t \}$ *be a set of non-zero polynomials in* $k[x_1, \ldots x_n]$. *Then* $G$ *is a Gröbner basis for* $I =< g_1, \ldots, g_t >$ *if and only if for all* $i \neq j$, *the remainder on division of* $S(f_i, f_j)$ *by* $G$ *(listed in some order ) is zero.*

**Proof** If $G = \{g_1, \ldots, g_t\}$ is a Gröbner basis for $I =< g_1, \ldots, g_t >$, then $\mathrm{rem}(S(g_i, g_j), G) = 0$, since $S(g_i, g_j) \in I$, where $\mathrm{rem}(S(g_i, g_j), G)$ denotes the remainder on division of $S(g_i, g_j)$ by $G$, Another notation for this is $\overline{S(g_i, g_j)}^G = r$.

Conversely, let's assume that $\mathrm{rem}(S(g_i, g_j), G) = 0$, for all $i \neq j$. Let $f \in I =< g_1, \ldots, g_t >$. We have to show that $\mathrm{LM}(f) \in < \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_t) >$. Since $f \in I$, we can write $f = h_1 g_1 + \cdots + h_t g_t$ for some $h_i \in k[x_1, \ldots, x_n]$. Let $X := \max_{1 \leq i \leq t}(\mathrm{LM}(h_i g_i))$, we choose $X$ to be the smallest over all representation of $f := \sum_\nu h_\nu g_\nu$ by the well orderings property i.e. Lemma 4.10. It is clear that $X := \max_{1 \leq i \leq t}(\mathrm{LM}(h_i g_i)) \succ \mathrm{LM}(f)$, or $X := \max[(\mathrm{LM}(h_i g_i),\ i = 1, \ldots, t)] = \mathrm{LM}(f)$. If $\max(\mathrm{LM}(h_i g_i),\ i = 1, \ldots, t) = \mathrm{LM}(f)$, we are done. Since $\mathrm{LM}(f) = \mathrm{LM}(h_i g_i) = \mathrm{LM}(h_i)\,\mathrm{LM}(g_i)$ for some $i$, hence $\mathrm{LM}(f) \in < \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_t) >$. Now assume that $X := \max[\mathrm{LM}(h_i g_i),\ i = 1, \ldots, t] \succ \mathrm{LM}(f)$. Let $\mathcal{S} = \{ i \mid \mathrm{LM}(h_i g_i) = X \} = \{ i_1, \ldots, i_k \}$. For $i \in \mathcal{S}$. Write $h_i = c_i X_i +$ lower terms. Set $g = \sum_{i \in \mathcal{S}} c_i X_i g_i$, then $\mathrm{LM}(X_i g_i) = X$ for all $i \in \mathcal{S}$. Since $f = g +$ lower terms, we have $\mathrm{LM}(g) = \mathrm{LM}(f) \lneq X$. By lemma 4.33, there exists $d_{ij} \in k$ such that

$$g = \sum_{i,j \in \mathcal{S}, i \neq j} d_{ij} S(X_i g_i, X_j g_j).$$

Now, $X = \mathrm{LCM}(\mathrm{LM}(X_i g_i), \mathrm{LM}(X_j g_j))$ ( in fact $X = \mathrm{LM}(X_i g_i)$ for all $i \in \mathcal{S}$), so

$$\begin{aligned} S(X_i g_i, X_j g_j) &= \frac{X}{\mathrm{LT}(X_i g_i)} X_i g_i - \frac{X}{\mathrm{LT}(X_j g_j)} X_j g_j \\ &= \frac{X}{\mathrm{LT}(g_i)} g_i - \frac{X}{\mathrm{LT}(g_j)} g_j = \frac{X}{X_{ij}} S(g_i, g_j) \end{aligned}$$

where $X_{ij} = \mathrm{LCM}(\mathrm{LM}(g_i), \mathrm{LM}(g_j))$. By hypothesis, $\mathrm{rem}(S(g_i, g_j), G) = 0$. Thus $\mathrm{rem}(S(X_i g_i, X_j g_j), G) = 0$, so $S(X_i g_i, X_j g_j) \in I$ by Corollary 4.29. This gives a representation: $S(X_i g_i, X_j g_j) = \sum_{\nu=1}^{t} h_{ij\nu} g_\nu$. where, by Theorem 4.13

$$\begin{aligned} \max_{1 \leq \nu \leq t} [(\mathrm{LM}(h_{ij\nu}) \, \mathrm{LM}(g_\nu))] &\leq \mathrm{LM}(S(X_i g_i, X_j g_j)) \\ &< \max[(\mathrm{LM}(X_i g_i), \mathrm{LM}(X_j g_j))] = X. \end{aligned}$$

Substituting these expressions into $g$ , and $g$ into $f$, we get $f = \sum_{i=1}^{t} h'_i g_i$, which is another representation of $f$ with

$$X' = \max_{1 \leq \nu \leq t} (\mathrm{LM}(h'_i g_i)) \lneq X = \max_{1 \leq i \leq t} [(\mathrm{LM}(h_i g_i))].$$

This is a contradiction with the hypothesis that $X$ was minimum over all representations of $f$. □

**Corollary 4.35** *Let* $G = \{g_1, \ldots, g_t\}$ *with* $g_i \neq 0, (1 \leq i \leq t)$. *Then* $G$ *is a Gröbner basis for* $I =< g_1, \ldots, g_t >$ *if and only if* $S(g_i, g_j) = \sum_{\nu=1}^{t} h_{ij\nu} g_\nu$, *where* $\mathrm{LM}(S(g_i, g_j)) = \max_{1 \leq \nu \leq t} [\mathrm{LM}(h_{ij\nu}) \, \mathrm{LM}(g_\nu)]$

4.2.7.1 *Buchberger's algorithm.* Buchberger's theorem above gives us a strategy for computing Gröebner bases: reduce(divide) the $S$-Polynomials (by $G$), and if a remainder is non-zero, add this remainder to the list of polynomials in the generating set($G$). Calculate all the $S$-polynomials which have not yet been calculated, and reduce them with respect to the new generating set, and do this until there are enough polynomials to make all $S$-polynomials reduce to zero. Note that, each time we extend the generating set, we also extend the monomial ideal generated by the leading terms of generators ($< \mathrm{LT}(g), g \in G >$ .) Thus termination is guaranteed by Hilbert's Basis Theorem: every ascending chain of Ideals in a Noetherian ring (here:$k[x_1, \ldots, x_n]$ ) stabilizes, and the final result is a Gröbner basis, by Buchberger theorem.

**Example 4.36** Consider the ring $k[x, y]$ with the Deglex (grelex) ordering, and let $I =< f_1, f_2 >=< x^3 - 2xy, x^2y - 2y^2 + x >$. Recall that from Example 4.21 $\{f_1, f_2\}$, is not a Gröbner basis for $I$ since $\mathrm{LT}(S(f_1, f_2)) = -x^2 \notin < \mathrm{LT}(f_1), \mathrm{LT}(f_2) >$. So $\overline{S(f_1, f_2)}^{\{f_1, f_2\}} := \mathrm{rem}(S(f_1, f_2), \{f_1, f_2\}) = -x^2 \neq 0$.

If we add $f_3 = -x^2$, then the previous remainder becomes :

$$\begin{aligned} \overline{S(f_1, f_2)}^{\{f_1, f_2, f_3\}} &= 0, \\ S(f_1, f_3) &= \frac{x^3}{x^3}(x^3 - 2xy) - \frac{x^3}{-x^2}(-x^2) = -2xy, \text{ and so} \\ \overline{S(f_1, f_3)}^{\{f_1, f_2, f_3\}} &= -2xy \neq 0. \end{aligned}$$

Hence we must add $f_4 = -2xy$ to our generating set. So we get

$$\begin{aligned}
\overline{S(f_1,f_2)}^{\{f_1,f_2,f_3,f_4\}} &= \overline{S(f_1,f_3)}^{\{f_1,f_2,f_3,f_4\}} = 0 \\
S(f_1,f_4) &= \frac{x^3y}{x^3}(x^3-2xy) - \frac{x^3y}{-2xy}(-2xy) = -2xy^2 = yf_4, \text{so} \\
\overline{S(f_1,f_4)}^{\{f_1,f_2,f_3,f_4\}} &= 0, \\
S(f_2,f_3) &= (x^2y-2y^2+x)-(-y)(-x^2) = -2y^2+x, \text{but} \\
\overline{S(f_2,f_3)}^{\{f_1,f_2,f_3,f_4\}} &= -2y^2+x \neq 0.
\end{aligned}$$

So we need to add $f_5 = \overline{S(f_2,f_3)}^{\{f_1,f_2,f_3,f_4\}} = -2y^2+x$ to the generating family. This time we get:

$$\overline{S(f_i,f_j)}^{\{f_1,f_2,f_3,f_4,f_5\}} = 0 \text{ for all } 1 \le i \le j \le 5.$$

**Theorem 4.37** *Let $I =< f_1,\ldots,f_s >\neq \{0\}$ be a polynomial ideal. Then a Gröbner basis can be constructed in a finite number of steps by the following algorithm:*

Input : $F = (f_1,\ldots,f_s)$

Output:a Gröbner basis $G = (g_1,\ldots,g_t)$ for $I$, with $F \subset G$

$G := F$

**REPEAT**

  $G' := G$

  **FOR** each pair $p, q, p \neq q \in G'$ **DO**

    $S := \overline{S(p,q)}^{G'}$

    **IF** $S \neq 0$ **THEN** $G := G \cup S$

**UNTIL** $G = G'$.

As mentioned before, this algorithm terminates ($k[x_1,\ldots,x_n]$ being a Noetherian ring). And it does when $G = G'$, where each pass through the main loop, $G$ consists of $G'$(the old $G$)$\cup \overline{S(p,q)}^{G'}$. So we have $\overline{S(p,q)}^{G'} = 0$ for all $p, q \in G$ (when $G = G'$). Hence $G$ is a Gröbner for $I$ by Theorem 4.34.

**4.2.8 Reduced Gröbner bases.** Gröbner bases are not unique. For any Gröebner basis $G$ of $I$, and any element $f \in I$, we can extend $G$ with $f$ and get a new Gröbner basis for $I$.

If $I =< g_1,\ldots,g_s >$ and $G = \{g_1,\ldots,g_s\}$ is a Gröbner basis, we define $G$ to be reduced if :

- $\{\mathrm{LT}(g_1),\ldots,\mathrm{LT}(g_s)\}$ constitutes a minimal set of generators for $\mathrm{LT}(I)$.
- $g_i$ are monic, i.e. $\mathrm{LC}(g_i) = 1$, $i = 1,\ldots,s$.
- No $\mathrm{LT}(g_i)$ divides any monomial of $g_j$, $i \neq j$.

$G$ is called a minimal Gröbner basis if it satisfies the first two conditions only.

**Example 4.38** For the ideal $I$ studied in the previous example, using the Deglex order, we found the Gröbner basis

$$f_1 = x^3 - 2xy$$
$$f_2 = x^2y - 2y^2 + x$$
$$f_3 = -x^2$$
$$f_4 = -2xy$$
$$f_5 = -2y^2 + x.$$

Note that $\mathrm{LT}(f_1) = x^3 = -x\,\mathrm{LT}(f_3)$, so we can remove $f_1$ ( i.e remove $\mathrm{LT}(f_1)$ from generators of $< \mathrm{LT}(I) >$). Similarly $\mathrm{LT}(f_2) = x^2y = (\frac{-1}{2}x)\,\mathrm{LT}(f_4)$, so $f_2$ can be removed too. Thus we can take $f_3' = x^2$, $f_4' = xy$, $f_5' = y^2 - \frac{1}{2}x$ as a minimal Gröbner basis for $I$ (The first two conditions above being satisfied). But $\hat{f}_3 = x^2 + axy$, $\hat{f}_4 = f_4' = xy$, $\hat{f}_5 = f_5' = y^2 - \frac{1}{2}x$, is also minimal in the sense of the above first two conditions. But we can see that $\mathrm{LT}(\hat{f}_4) = xy$ divides $axy$ which is a monomial of $\hat{f}_3$. So unlike the previous family, this one is not reduced.

**Proposition 4.39** *Every non-zero polynomial ideal $I$, has a unique reduced Gröbner basis for a given monomial ordering.*

**Proof** We give only a sketch of the proof here. We do however, refer the reader for a more rigorous proof and a more elaborate text on the subject, to the book [21], on which we have based most of the material covered in this article.
Given a Gröbner basis $G = \{g_1, \ldots, g_s\}$, we can always render it minimal (by removing redundant elements and dividing by the leading coefficients), say $G' = \{g_{i1}, \ldots, g_{ik}\}$. We then take the remainder $\overline{g_{ik}}^{G'-\{g_{ik}\}}$ on the division of $g_{ik}$ by $G' - \{g_{ik}\}$ and thus for $i_1, \ldots i_k$ successively. The result is a reduced Gröbner basis. Given two(distinct) reduced Gröbner bases $G = \{g_1, \ldots, g_s\}$, and $H = \{h_1, \ldots, h_s\}$ (by Minimality, they must have the same Cardinality) for the same ideal $I$. Given any $g_i$, there exists $h_k$, which for simplicity we call it $h_i$ such that $\mathrm{LT}(g_i) = \mathrm{LT}(h_i)$ (we can always reindex $h_i$'s for such a purpose) . These two terms will be canceled in $g_i - h_i$, the remaining monomials are not divisible by any $\mathrm{LT}(g_j) = \mathrm{LT}(h_j)$ for any $j \neq i$ (reduced bases), this contradicts the fact that $g_i - h_i \in I$. So $g_i = h_i$. □

**Mohammed Radi-Benjelloun**
1612 Dundas St. W., Apt. 6
Toronto, ON, Canada M6K 1T8
radi97@hotmail.com



Fields Institute Communications
Volume **58**, 2010

CHAPTER 5

# $C_{ab}$ Curves and Arithmetic on Their Jacobians

FARZALI IZADI

**Abstract.** In this expository report, we first introduce the $C_{ab}$ curves, which includes the class of elliptic, hyperelliptic and even superelliptic curves. Then we construct an efficient algorithm for addition in the Jacobian group of $C_{ab}$ curves. The main idea is to represent any element in the ideal class group by its unique reduced Gröbner basis.

## 5.1 Introduction

Suppose a class $C$ of algebraic curves is given. To construct a discrete-log based cryptosystem using the class $C$, we need to solve the following two basic problems.

**Problem 1**. Give an efficient algorithm for addition in the Jacobian group of any curve in the class $C$.

**Problem 2**. Give an efficient algorithm to find a curve with a Jacobian of almost prime order in the class $C$.

The above two problems have been solved for elliptic curves. For hyperelliptic curves Problem 1 is solved in Cantor [17], Koblitz [51], Lange [55], Guyot-Kaveh-Patankar [40], Wollinger-Pelzl-Paar [88], and Problem 2 is partially solved in Koblitz [52], Gaudry [34], Matsuda-Chao-Tsujii [60], Izadi-Murty [45], Gaudry-Harley [35], Harley [41], Spallek [80].

In [64] Miura-Kamiya, have found a class of algebraic curves called $C_{ab}$ curves in their work on algebraic geometry codes. This includes elliptic and hyperelliptic curves. Moreover, $C_{ab}$ curves have good properties to facilitate computations.

In this paper, we first characterize the defining equations of a $C_{ab}$ curves completely, then we solve Problem 1 for $C_{ab}$ curves, that is, to construct an efficient algorithm for addition in the Jacobian of $C_{ab}$ curves, thus showing the possibility of a discrete-log based cryptosystems using $C_{ab}$ curves. In the end, we will work out some examples from elliptic, hyperelliptic, and also superelliptic cases.

## 5.2 Preliminaries

We arrange facts about algebraic function fields, divisor class group of a function field, subrings of function fields (strictly speaking a holomorphy ring of the function field), Jacobian group of an algebraic curve and the Gröbner basis of an ideal in a polynomial ring.

### 5.2.1 Algebraic function fields.

**Definition 5.1** An *algebraic function field $F/K$ of one variable over $K$* is an extension field $K \subseteq F$ such that $F$ is a finite algebraic extension of $K(x)$ for some element $x \in F$ which is transcendental over $K$.

For brevity, we shall simply refer to $F/K$ as a *function field.* The set $\tilde{K} = \{z \in F | z$ is algebraic over $K\}$ is a subfield of $F$. We have $K \subseteq \tilde{K} \subset F$, and also $F/\tilde{K}$ is a function field over $\tilde{K}$. We say that $K$ is *algebraically closed in* $F$, or $K$ is the full constant field of $F$ if $\tilde{K} = K$.

In order to formulate, whether any element $0 \neq z \in F$ admits a decomposition into irreducibles, we introduce the notions of valuation rings and places.

**Definition 5.2** A *valuation ring* of the function field $F/K$ is a ring $\mathcal{O} \subseteq F$ with the following properties:

(a) $K \subsetneqq \mathcal{O} \subsetneqq F$, and
(b) for any $z \in F, z \in \mathcal{O}$ or $z^{-1} \in \mathcal{O}$.

The following proposition is immediate.

**Proposition 5.3** *Let $\mathcal{O}$ be a valuation ring of the function field $F/K$. Then*

(a) *$\mathcal{O}$ is a local ring, i.e., $\mathcal{O}$ has a unique maximal ideal $P = \mathcal{O}\backslash\mathcal{O}^*$, where $\mathcal{O}^* = \{z \in \mathcal{O} | \text{there is a } w \in \mathcal{O} \text{ with } zw = 1\}$ is the group of units of $\mathcal{O}$.*
(b) *For $0 \neq x \in F$, $x \in P \Longleftrightarrow x^{-1} \notin \mathcal{O}$.*
(c) *For the field $\tilde{K}$ of constants of $F/K$ we have $\tilde{K} \subseteq \mathcal{O}$ and $\tilde{K} \cap P = \{0\}$.*

**Theorem 5.4** *Let $\mathcal{O}$ be a valuation ring of the function field $F/K$ and $P$ be its unique maximal ideal. Then*

(a) *$P$ is a principal ideal.*
(b) *If $P = t\mathcal{O}$ then any $0 \neq z \in F$ has a unique representation of the form $z = t^n u$ for some $n \in \mathbb{Z}$, $u \in \mathcal{O}^*$.*
(c) *$\mathcal{O}$ is a principal ideal domain. More precisely, if $P = t\mathcal{O}$ and $\{0\} \neq I \subseteq \mathcal{O}$ is an ideal then $I = t^n\mathcal{O}$ for some $n \in \mathbb{N}$.*

A ring having the above properties is called a *discrete valuation ring.*

**Definition 5.5**

(a) A *place* $P$ of the function field $F/K$ is the maximal ideal of some valuation ring $\mathcal{O}$ of $F/K$. Any element $t \in P$ such that $P = t\mathcal{O}$ is called a *prime element* for $P$ (other notations are *local parameter* or *uniformizing variable*).
(b) $\mathbb{P}_F := \{P | P \text{ is a place of } F/K\}$.

If $\mathcal{O}$ is a valuation ring of $F/K$ and $P$ its maximal ideal, then $\mathcal{O}$ is uniquely determined by $P$, namely $\mathcal{O} = \{z \in F | z^{-1} \notin P\}$, cf. Proposition 5.3(b). Hence $\mathcal{O}_P := \mathcal{O}$ is called the *valuation ring of the place $P$.*

A second useful description of places is given in terms of valuations.

**Definition 5.6** *A discrete valuation* of $F/K$ is a function $v : F \to \mathbb{Z} \cup \{\infty\}$ with the following properties:

(a) $v(x) = \infty \Longleftrightarrow x = 0$.
(b) $v(xy) = v(x) + v(y)$ for any $x, y \in F$.
(c) $v(x + y) \geq \min\{v(x), v(y)\}$ for any $x, y \in F$.
(d) There exists an element $z \in F$ with $v(z) = 1$.
(e) $v(a) = 0$ for any $0 \neq a \in K$.

In this context, the symbol $\infty$ means some element not in $\mathbb{Z}$ such that $\infty+\infty = \infty + n = n + \infty = \infty$ and $\infty > m$ for all $m, n \in \mathbb{Z}$. From (b) and (d) it follows immediately that $v : F \to \mathbb{Z} \cup \{\infty\}$ is surjective. Property (c) is called the *Triangle Inequality*.

The next definition together with the following theorem reveals the relationship between places and discrete valuations.

**Definition 5.7** To any place $P \in \mathbb{P}_F$ we associate a function $v_P : F \to \mathbb{Z} \cup \{\infty\}$ that will prove to be a discrete valuation of $F/K$: Choose a prime element $t$ for $P$. Then every $0 \neq z \in F$ has a unique representation $z = t^n u$ with $u \in \mathcal{O}_P^*$ and $n \in \mathbb{Z}$. Define $v_P(z) := n$ and $v_P(0) := \infty$.

Observe that this definition depends only on $P$, not on the choice of $t$. In fact, if $t'$ is another prime element for $P$ then $P = t\mathcal{O} = t'\mathcal{O}$, so $t = t'w$ for some $w \in \mathcal{O}_P^*$. Therefore $t^n u = (t'^n w^n)\, u = t'^n\,(w^n u)$ with $w^n u \in \mathcal{O}_P^*$.

**Theorem 5.8** *Let $F/K$ be a function field.*

(a) *For any place $P \in \mathbb{P}_F$, the function $v_P$ defined above is a discrete valuation of $F/K$. Moreover, we have*

$$\mathcal{O}_P = \{z \in F | v_P(z) \geq 0\}\ ,$$
$$\mathcal{O}_P^* = \{z \in F | v_P(z) = 0\}\ ,$$
$$P = \{z \in F | v_P(z) > 0\}\ .$$

*An element $x \in F$ is a prime element for $P$ if and only if $v_P(x) = 1$.*

(b) *Conversely, suppose that $v$ is a discrete valuation of $F/K$. Then the set $P = \{z \in F | v(z) > 0\}$ is a place of $F/K$, and $\mathcal{O}_P = \{z \in F | v(z) \geq 0\}$ is the corresponding valuation ring.*

(c) *Any valuation ring $\mathcal{O}$ of $F/K$ is a maximal proper subring of $F$.*

According to Theorem 5.8, places, valuation rings and discrete valuations of a function field essentially amount to the same thing.

Let $P$ be a place of $F/K$, and $\mathcal{O}_P$ be its valuation ring. Since $P$ is a maximal ideal, the residue class ring $\mathcal{O}_P/P$ is a field. For $x \in \mathcal{O}_P$ we define $x(P) \in \mathcal{O}_P/P$ to be the residue class of $x$ modulo $P$, for $x \in F \setminus \mathcal{O}_P$ we put $x(P) := \infty$ (note that the symbol $\infty$ is used here in a different sense as in Definition 5.6). By Proposition 5.3(c), we know that $K \subseteq \mathcal{O}_P$ and $K \cap P = \{0\}$, so the residue class map $\mathcal{O}_P \to \mathcal{O}_P/P$ induces a canonical embedding of $K$ into $\mathcal{O}_P/P$. Henceforth we shall always consider $K$ as a subfield of $\mathcal{O}_P/P$ via this embedding. Observe that this argument also applies to $\widetilde{K}$ instead of $K$; so we can consider $\widetilde{K}$ as a subfield of $\mathcal{O}_P/P$ as well.

**Definition 5.9** Let $P \in \mathbb{P}_F$.

(a) $F_P := \mathcal{O}_P/P$ is the *residue class field* of $P$. The map $x \mapsto x(P)$ from $F$ to $F_P \cup \{\infty\}$ is called the *residue class map* with respect to $P$. Sometimes we shall also use the notation $x + P := x(P)$ for $x \in \mathcal{O}_P$.

(b) $\deg P := [F_P : K]$ is called the *degree* of $P$.

The degree of a place is always finite; more precisely the following holds.

**Proposition 5.10** *If $P$ is a place of $F/K$ and $0 \neq x \in P$ then*

$$\deg P \leq [F : K(x)] < \infty\ .$$

**Corollary 5.11** *The field* $\widetilde{K}$ *of constants of* $F/K$ *is a finite field extension of* $K$.

**Remark 5.12** For the case when $\deg P = 1$ we have $F_P = K$, and the residue class map maps $F$ to $K \cup \{\infty\}$. In particular, if $K$ is an algebraically closed field, any place has degree one, so we can read an element $z \in F$ as a function

$$z : \begin{cases} \mathbb{P}_F \to K \cup \{\infty\} , \\ P \mapsto z(P) . \end{cases} \tag{5.1}$$

This is why $F/K$ is called a *function field*. The elements of $K$, interpreted as functions in the sense of the map 5.1, are constant functions. For this reason $K$ is called the *constant field* of $F$. Also, the following terminology is justified by 5.1:

**Definition 5.13** Let $z \in F$ and $P \in \mathbb{P}_F$. We say that $P$ is a *zero* of $z$ iff $v_P(z) > 0$; $P$ is a *pole* of $z$ iff $v_P(z) < 0$. If $v_P(z) = m > 0$, $P$ is a *zero of $z$ of order $m$*; if $v_P(z) = -m < 0$, $P$ is a *pole of $z$ of order $m$*.

Next we shall be concerned with the question as to whether there exist places of $F/K$.

**Theorem 5.14** *Let* $F/K$ *be a function field and* $R$ *be a subring of* $F$ *with* $K \subseteq R \subseteq F$. *Suppose that* $\{0\} \neq I \subsetneqq R$ *is a proper ideal of* $R$. *Then there is a place* $P \in \mathbb{P}_F$ *such that* $I \subseteq P$ *and* $R \subseteq \mathcal{O}_P$.

**Corollary 5.15** *Let* $F/K$ *be a function field,* $z \in F$ *transcendental over* $K$. *Then* $z$ *has at least one zero and one pole. In particular,* $\mathbb{P}_F \neq \emptyset$.

**5.2.2 Independence of valuations.** The main result of this section is the *Weak Approximation Theorem* 5.16 (which is also referred to as the *Theorem of Independence*). Essentially this says the following: If $v_1, \ldots, v_n$ are pairwise distinct valuations of $F/K$ and $z \in F$, and if we know the values $v_1(z), \ldots, v_{n-1}(z)$, then we cannot conclude anything about $v_n(z)$.

**Theorem 5.16 (Weak Approximation Theorem)** *Let* $F/K$ *be a function field,* $P_1, \ldots, P_n \in \mathbb{P}_F$ *pairwise distinct places of* $F/K$, $x_1, \ldots, x_n \in F$ *and* $r_1, \ldots, r_n \in \mathbb{Z}$. *Then there is some* $x \in F$ *such that* $v_P(x - x_i) = r_i$.

**Corollary 5.17** *Any function field has infinitely many places.*

**Proposition 5.18** *Let* $F/K$ *be a function field and* $P_1, \ldots, P_r$ *be zeros of the element* $x \in F$. *then*

$$\sum_{i=1}^{r} v_{P_i}(x) \cdot \deg P_i \leq [F : K(x)] \ .$$

**Corollary 5.19** *In a function field* $F/K$, *any element* $0 \neq x \in F$ *has only finitely many zeros and poles.*

**5.2.3 Divisors of function fields.**

**Definition 5.20** The (additively written) free abelian group which is generated by the places of $F/K$ is denoted by $\mathcal{D}_F$, the *divisor group* of $F/K$. The elements of $\mathcal{D}_F$ are called *divisors* of $F/K$. In other words, a divisor is a formal sum

$$D = \sum_{P \in \mathbb{P}_F} n_P P \text{ with } n_P \in \mathbb{Z}, \text{ almost all } n_P = 0.$$

A divisor of the form $D = P$ with $P \in \mathbb{P}_F$ is called a *prime divisor.* Two divisors are added coefficientwise. The zero element of the divisor group $\mathcal{D}_F$ is the divisor 0. For $Q \in \mathbb{P}_F$ and $D = \sum n_P P \in \mathcal{D}_F$ we define $v_Q(D) = n_Q$, therefore

$$D = \sum v_P(D).P.$$

A partial ordering on $\mathcal{D}_F$ is defined by

$$D_1 \leq D_2 \text{ iff } v_P(D_1) \leq v_P(D_2) \text{ for any } P \in \mathbb{P}_F.$$

A divisor $D \geq 0$ is called *positive* or *effective.* The *degree* of a divisor is defined by

$$\deg(D) = \sum_{P \in \mathbb{P}_F} v_P(D). \deg P$$

and yields a homomorphism $\deg : \mathcal{D}_F \to \mathbb{Z}$.

By Corollary 5.19, any nonzero element $x \in F$ has only finitely many zeros and poles in $\mathbb{P}_F$. Thus the following definition makes sense.

**Definition 5.21** Let $0 \neq x \in F$ and denote by $Z$ (resp. $N$) the set of zeros (poles) of $x$ in $\mathbb{P}_F$. Then we define

$$\begin{aligned} (x)_0 &:= \sum_{P \in Z} v_P(x)P, \text{ the } \textit{zero divisor} \text{ of } x, \\ (x)_\infty &:= \sum_{P \in N} (-v_P(x))P, \text{ the } \textit{pole divisor} \text{ of } x, \\ (x) &:= (x)_0 - (x)_\infty, \text{ the } \textit{principal divisor} \text{ of } x. \end{aligned}$$

Clearly $(x)_0 \geq 0, (x)_\infty \neq 0$ and

$$(x) = \sum_{P \in \mathbb{P}_F} v_P(x)P.$$

The set $\mathbb{P}_F := \{(x) | 0 \neq x \in F\}$ is called the *group of principal divisors* of $F/K$. This is a subgroup of $\mathcal{D}_F$. The factor group

$$\mathcal{C} = \mathcal{D}_F / \mathcal{P}$$

is the *divisor class group.*

For a divisor $A \in \mathcal{D}_F$ we set

$$\mathcal{L}(A) := \{x \in F | (x) \geq -A\} \cup \{0\}.$$

Then we have

**Proposition 5.22** *The space $\mathcal{L}(A)$ is a finite-dimensional vector space. More precisely: If $A = A_+ - A_-$ with positive divisors $A_+$ and $A_-$, then*

$$\dim \mathcal{L}(A) \leq \deg A_+ + 1.$$

**Theorem 5.23** *Any principal divisor has degree zero. More precisely: Let $x \in F \setminus K$ and $(x)_0$ resp. $(x)_\infty$ denote the zero resp. pole divisor of $x$. Then*

$$\deg(x)_0 = \deg(x)_\infty = [F : K(x)] \ .$$

**Proof of the Theorem.** Set $n := [F : K(x)]$ and

$$B := (x)_\infty = \sum_{i=1}^{r} -v_{P_i}(x) P_i \ ,$$

where $P_1, \ldots, P_r$ are all the poles of $x$. Then

$$\deg B = \sum_{i=1}^{r} v_{P_i}\left(x^{-1}\right) \cdot \deg P_i \leq [F : K(x)] = n$$

by Proposition 5.18, and thus it remains to show that $n \leq \deg B$ as well. Choose a basis $u_1, \ldots, u_n$ of $F/K(x)$ and a divisor $C \geq 0$ such that $(u_i) \geq -C$ for $i = 1, \ldots, n$. We have

$$\dim(lB + C) \geq n(l+1) \quad \text{for all } l \geq 0 \; , \tag{5.2}$$

which follows immediately from the fact that $x^i u_j \in \mathcal{L}(lB + C)$ for $0 \leq i \leq l$, $1 \leq j \leq n$ (observe that these elements are linearly independent over $K$ since $u_1, \ldots, u_n$ are linearly independent over $K(x)$). Setting $c := \deg C$ we obtain $n(l+1) \leq \dim(lB + C) \leq l \cdot \deg B + c + 1$ by Proposition 5.22. Thus

$$l(\deg B - n) \geq n - c - 1 \tag{5.3}$$

for all $l \in \mathbb{N}$. The right hand side of (5.3) is independent of $l$, therefore (5.3) is possible only when $\deg B \geq n$.

We have proved that $\deg(x)_\infty = [F : K(x)]$. Since $(x)_0 = \left(x^{-1}\right)_\infty$, we conclude that $\deg(x)_0 = \deg\left(x^{-1}\right)_\infty = \left[F : K\left(x^{-1}\right)\right] = [F : K(x)]$. □

**5.2.4 Subrings of function fields.** *As before, $F/K$ denotes a function field with constant field $K$.*

**Definition 5.24** A *subring* of $F/K$ is a ring $R$ such that $K \subseteq R \subseteq F$, and $R$ is not a field.

In particular, if $R$ is a subring of $F/K$ then $K \subsetneqq R \subsetneqq F$. Here are two typical examples:

(a) $R = \mathcal{O}_P$ for some $P \in \mathbb{P}_F$.
(b) $R = K[x_1, \ldots, x_n]$ where $x_1, \ldots, x_n \in F \setminus K$.

While $\mathcal{O}_P$ is obviously a subring, to see that $K[x_1, \ldots, x_n]$ is also a subring, we have to show that it is not a field. To this end, choose a place $P \in \mathbb{P}_F$ such that $v_P(x_1) \geq 0, \ldots, v_P(x_n) \geq 0$. Let $x = x_1$ and $d := \deg P$. As the residue classes $1, x(P), \ldots, x^d(P) \in \mathcal{O}_P/P$ are linearly dependent over $K$ one can find $\alpha_0, \ldots, \alpha_d \in K$ such that the element $z = \alpha_0 + \alpha_1 x + \cdots + \alpha_d x^d$ is not 0 but $v_P(z) > 0$ (observe that $x$ is transcendental over $K$ since $x \notin K$). Clearly $z \in K[x_1, \ldots, x_n]$ but $z^{-1} \notin K[x_1, \ldots, x_n]$ (since $v_P(y) \geq 0$ for any $y \in K[x_1, \ldots, x_n]$).

A more general example than (a) is given by the following definition.

**Definition 5.25** For $\emptyset \neq S \subsetneqq \mathbb{P}_F$ let

$$\mathcal{O}_S := \{z \in F | v_P(z) \geq 0 \text{ for all } P \in S\}$$

be the intersection of all valuation rings $\mathcal{O}_P$ with $P \in S$. Any ring $R \subseteq F$ which is of the form $R = \mathcal{O}_S$ for some $S \subsetneqq \mathbb{P}_F$, $S \neq \emptyset$ is called a *holomorphy ring* of $F/K$.

We note some simple consequences.

**Lemma 5.26**

(a) *Any valuation ring $\mathcal{O}_P$ is a holomorphy ring, namely $\mathcal{O}_P = \mathcal{O}_S$ with $S = \{P\}$.*
(b) *Any holomorphy ring $\mathcal{O}_S$ is a subring of $F/K$.*

(c) *For* $P \in \mathbb{P}_F$ *and* $\emptyset \neq S \subsetneqq \mathbb{P}_F$ *we have*

$$\mathcal{O}_S \subseteq \mathcal{O}_P \Longleftrightarrow P \in S \ .$$

*Consequently,* $\mathcal{O}_S = \mathcal{O}_T \Leftrightarrow S = T$.

For the proof of this fact, we need the following Strong Approximation Theorem.

**Strong Approximation Theorem.** *Let* $S \subsetneqq \mathbb{P}_F$ *be a proper subset of* $\mathbb{P}_F$ *and* $P_1, P_2, ..., P_n \in S$. *Suppose there are given* $x_1, x_2, ..., x_n \in F$ *and* $r_1, r_2, ..., r_n \in \mathbb{Z}$. *Then there exists an element* $x \in F$ *such that* $v_P(x - x_i) = r_i, (i = 1, 2, ..., n)$ *and* $v_P(x) \geq 0$ *for all* $P \in \{P_1, P_2, ..., P_n\}$.

**Proof** (b) Since $\mathcal{O}_S$ is a ring with $K \subseteq \mathcal{O}_S \subseteq F$ we have only to show that it is not a field. Choose a place $P_1 \in S$. As $S \neq \mathbb{P}_F$, the Strong Approximation Theorem yields an element $0 \neq x \in F$ such that

$$v_{P_1}(x) > 0 \qquad \text{and} \qquad v_P(x) \geq 0 \qquad \text{for all } P \in S \ .$$

Obviously $x \in \mathcal{O}_S$ but $x^{-1} \notin \mathcal{O}_S$; therefore $\mathcal{O}_S$ is not a field.

(c) Suppose $P \notin S$. By the Strong Approximation Theorem, we can find $z \in F$ with

$$v_P(z) < 0 \qquad \text{and} \qquad v_Q(z) \geq 0 \qquad \text{for any } Q \in S. \tag{5.4}$$

(This is clear if $S \cup \{P\} \neq \mathbb{P}_F$. If however $S \cup \{P\} = \mathbb{P}_F$, choose $z \in \mathcal{O}_S$ which has at least one zero in $S$; since $z$ must have some pole it follows that $v_P(z) < 0$.) Any element $z$ satisfying (5.4) is in $\mathcal{O}_S$ but not in $\mathcal{O}_P$. Thus we have proved that $P \notin S$ implies $\mathcal{O}_S \nsubseteq \mathcal{O}_P$. The remaining assertions are trivial. □

**Definition 5.27** Let $R$ be a subring of $F/K$.

(a) An element $z \in F$ is said to be *integral over* $R$ if $f(z) = 0$ for some monic polynomial $f(X) \in R[X]$, i.e. if there are $a_0, \ldots, a_{n-1} \in R$ such that

$$z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = 0 \ .$$

Such an equation is called an *integral equation* for $z$ over $R$.

(b) The set

$$ic_F(R) := \{z \in F | z \text{ is integral over } R\}$$

is called the *integral closure* of $R$ in $F$.

(c) Let $F_0 \subseteq F$ denote the quotient field of $R$. The ring is called *integrally closed* if $ic_{F_0}(R) = R$, i.e. any $z \in F_0$ which is integral over $R$ is already in $R$.

**Proposition 5.28** *Let* $\mathcal{O}_S$ *be a holomorphy ring of* $F/K$. *Then*

(a) $F$ *is the quotient field of* $\mathcal{O}_S$.

(b) $\mathcal{O}_S$ *is integrally closed.*

**Proof** (a) Let $x \in F$, $x \neq 0$. By the Strong Approximation Theorem, there is an element $z \in F$ such that

$$v_P(z) \geq \max\{0, v_P(x^{-1})\} \quad \text{for all } P \in S$$

.

Clearly $z \in \mathcal{O}_S$ and $y := zx \in \mathcal{O}_S$, so $x = yz^{-1}$ is in the quotient field of $\mathcal{O}_S$.

(b) Let $u \in F$ be integral over $\mathcal{O}_S$. Choose an integral equation

$$u^n + a_{n-1}u^{n-1} + \cdots + a_0 = 0 \tag{5.5}$$

with $a_i \in \mathcal{O}_S$. We have to show that $v_P(u) \geq 0$ for all $P \in S$. Suppose this is false, so that $v_P(u) < 0$ for some $P \in S$. Since $v_P(a_i) \geq 0$,

$$v_P(u^n) = n \cdot v_P(u) < v_P(a_i u^i)$$

for $i = 0, \ldots, n-1$. Thus the Strict Triangle Inequality yields a contradiction to (5.5). □

**Theorem 5.29** *Let $R$ be a subring of $F/K$ and $S(R) := \{P \in \mathbb{P}_F | R \subseteq \mathcal{O}_P\}$ . Then the following holds:*

(a) $\emptyset \neq S(R) \subsetneqq \mathbb{P}_F$,

(b) *The integral closure of $R$ in $F$ is $ic_F(R) = \mathcal{O}_{S(R)}$. In particular, $ic_F(R)$ is an integrally closed subring of $F/K$ with quotient field $F$.*

**Proof** (a) Since $R$ is not a field we can find a proper ideal $I \subsetneqq R$, and by Theorem 5.14, there exists a place $P \in \mathbb{P}_F$ such that $I \subseteq P$ and $R \subseteq \mathcal{O}_P$. Therefore $S(R) \neq \emptyset$. On the other hand, consider an element $x \in R$ which is transcendental over $K$. Any place $Q \in \mathbb{P}_F$ which is a pole of $x$ is not in $S(R)$, so $S(R) \neq \mathbb{P}_F$.

(b) Since $R \subseteq \mathcal{O}_{S(R)}$ and $\mathcal{O}_{S(R)}$ is integrally closed (by Proposition 5.28) it follows immediately that $ic_F(R) \subseteq \mathcal{O}_{S(R)}$. In order to prove the inclusion in the reverse direction, consider an element $z \in \mathcal{O}_{S(R)}$. We claim:

$$z^{-1} \cdot R\left[z^{-1}\right] = R\left[z^{-1}\right] . \tag{5.6}$$

Suppose that (5.6) is false, i.e., $z^{-1} \cdot R\left[z^{-1}\right]$ is a proper ideal in $R\left[z^{-1}\right]$. By Theorem 5.14, we find a place $Q \in \mathbb{P}_F$ such that

$$R\left[z^{-1}\right] \subseteq \mathcal{O}_Q \quad \text{and} \quad z^{-1} \in Q .$$

It follows that $Q \in S(R)$ and $z \notin \mathcal{O}_Q$ which is a contradiction to $z \in \mathcal{O}_{S(R)}$; thus we have proved (5.6). From (5.6) we obtain a relation

$$1 = z^{-1} \cdot \sum_{i=0}^{s} a_i \left(z^{-1}\right)^i \tag{5.7}$$

with $a_0, \ldots, a_s \in R$. Multiplying (5.7) by $z^{s+1}$ yields

$$Z^{s+1} - \sum_{i=0}^{s} a_i z^{s-i} = 0.$$

This is an integral equation for $z$ over $R$. For detailed account of Sections 2.1–2.4, see Stichtenoth, [81]. □

### 5.2.5 Jacobian group of an algebraic curve.

**Definition 5.30** Take an algebraic curve $C$ defined over a finite field $K$. Let $\bar{K}$ be an algebraic closure of $K$. For integers $m_i$ and rational points (strictly, places) $P_i$ of $C$ over $\bar{K}$, a formal sum $D = \sum m_i P_i$ is called a divisor. A divisor $D = \sum m_i P_i$ is called positive if $m_i \geq 0$ for all $i$. The integer $m = \sum m_i$ is called the degree of divisor $D = \sum m_i P_i$, denoted by $deg(D)$. All divisors on a curve $C$ become an abelian group $\mathcal{D}$ under formal additions, and all divisors of degree zero is a subgroup $\mathcal{D}^0$ of $\mathcal{D}$. Let $\mathcal{D}_K$ and $\mathcal{D}_K^0$ be invariant subgroups of $\mathcal{D}$ and $\mathcal{D}^0$, respectively, under the action of $Gal(\bar{K}|K)$. Elements of $\mathcal{D}_K$ are called divisors defined over $K$. For a rational function $f$ on a curve $C$, let $v_P(f) = n$ (or,$-n$) be the order of $f$ at a point $P$. Then, $(f) := \sum_P v_P(f) P$ becomes a divisor of degree 0, is called a principal divisor of $f$. The partial sums $(f)_0 := \sum_{P, v_P(f) \geq 0} v_P(f) P$

and $(f)_\infty := \sum_{P,v_P(f)\leq 0} -v_P(f)P$ are called a zero divisor and a pole divisor of $f$, respectively. Note both $(f)_0$ and $(f)_\infty$ are positive divisors, and $(f) = (f)_0 - (f)_\infty$. All principal divisors $\{(f)|f \in \bar{K}(C)\}$ become a subgroup $\mathbb{P}$ of $\mathcal{D}^0$. The residue group of $\mathcal{D}^0$ by $\mathbb{P}$ is called Jacobian group of $C$, denoted by $\mathbb{J}(C)$. The invariant subgroup $\mathbb{J}_K(C)$ of $\mathbb{J}(C)$ under the action of $Gal(\bar{K}|K)$ is called Jacobian group of $C$ defined over $K$. For a divisor $D$ defined over $K$, as in the function field case,

$$\mathcal{L}(D) = \{f \in K(C)|(f) + D \geq 0\} \cup \{0\}$$

is a finite dimensional vector space over $K$. From Riemann's theorem, we see that $dim\mathcal{L}(D) \geq deg(D) + 1 - g$, where $g$ denotes the genus of $C$.

For details, see (Silverman [76]).

**5.2.6 Monomial order and Gröbner bases.** Let $Z \geq 0$ denotes the set of non-negative integers. For a $n$-variable monomial $x^\alpha = x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}$, a $n$-tuple of integers $\alpha = (\alpha_1, \ldots, \alpha_n) \in Z^n_{\geq 0}$ is called a multi-degree of $x^\alpha$, denoted by $MD(x^\alpha)$. A well-order $<$ on $Z^n_{\geq 0}$ is called a monomial order if $\alpha + \gamma < \beta + \gamma$ holds whenever $\alpha < \beta$ and $\gamma \in Z^n_{\geq 0}$. A monomial order on $z^n_{\geq 0}$ determines a well-order on the set of all monomials through multi-degrees, called a monomial order, too. Suppose a monomial order is given. For a $n$-variable polynomial $f$, the maximum term appearing in $f$ with respect to the monomial order is called a leading term of $f$, denoted by $LT(f)$. By using monomial orders, we can describe the division algorithm for $n$-variable polynomials, in which a polynomial $f$ is divided by a set of polynomials $G$.

The ideal generated by polynomials $g_1, \ldots, g_m$ is denoted by $\langle g_1, ..., g_m\rangle$. Fix a monomial order.

**A subset $G = \{g_1, ..., g_m\}$ of an ideal $I$ of $n$-variable polynomial ring $K[x_1, \ldots, x_n]$ is called a Gröbner basis of $I$ when we have**

$$\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_m)\rangle$$

.

The following are two fundamental facts regarding a Gröbner basis.

(1) *Any ideal of $n$-variable polynomial ring $K[x_1, \ldots, x_n]$ has a Gröbner basis.*
(2) *If $G = \{g_1, \ldots, g_m\}$ is a Gröbner basis of an ideal $I$, then $G$ generates $I$.*

For an ideal $I$, all of the multi-degrees of monomials outside $LT(I)$ is called $\Delta$-set of $I$, denoted by $\Delta(I)$:

$$\Delta(I) = \{\alpha \in Z^n_{\geq 0}|x^\alpha \notin LT(I)\}.$$

Let $\delta(I)$ denote the number of elements in $\Delta(I)$.

On the other hand, for a polynomial set $G = \{g_1, \ldots, g_m\}$, we set

$$\delta(g_1, ..., g_m) := Card(Z^n_{\geq 0} - \cup_{i=1}^m (MD(LT(g_i)) + Z^n_{\geq 0})).$$

(Card(S) denotes the number of elements in the set S.) Then, for an ideal $I$ satisfying $\delta(I) < \infty$ and for its subset $G = \{g_1, \ldots, g_m\}$, we have

$$G \text{ is a Gröbner basis of } I \Longleftrightarrow \delta(I) = \delta(g_1, \ldots, g_m). \qquad (5.8)$$

Use of a Gröbner basis justifies the division algorithm for $n$-variable polynomials. That is, a polynomial $f$ is a member of an ideal $I$ if and only if the remainder of $f$ divided by the Gröbner basis of $I$ is equal to zero. Although there are several

Gröbner bases for a given ideal, the reduced Gröbner basis is uniquely determined by a given monomial ideal.

*A Gröbner basis $G$ of an ideal $I$ is called reduced when*

(a) *The coefficient of $LT(p)$ is 1 for all $p \in G$,*
(b) *Any term appearing in $p$ doesn't belong to $\langle LT(G - \{p\}) \rangle$ for all $p \in G$.*

For details, see [Cox-Little-O'Shea [21]].

### 5.3 The $C_{ab}$ curves

In this section, we characterize the defining equation of a plane algebraic curve $C$ with a unique point $Q_\infty$ at infinity. This corresponds to a place $Q$ of degree one at infinity. This characterization can be regarded as a generalization of the Weierstrass form of a hyperelliptic curve. Throughout in this note, $K$ denotes a perfect field, and $a$, $b$ denote relatively prime positive integers. For a place $Q$ of degree one over $K$ ($K$-rational), we take the ring $\mathcal{L}(\infty Q)$ of functions on $C$ which are holomorphic away from $Q$:

$$\mathcal{L}(\infty Q) = \{f \in K(C) | v_P(f) \geq 0 (\text{for all } P \neq Q)\}.$$

where $v_Q$ denotes the discrete valuation at $Q$. All of the pole number $-v(f)$ at $Q$ of $f$ in $\mathcal{L}(\infty Q)$ become a monoid $M_Q$:

$$M_Q = \{-v(f) | f \in \mathcal{L}(\infty Q)\}.$$

**Definition 5.31** ($C_{ab}$**-curve**) When $M_Q$ is generated by two natural numbers $a$ and $b$, we call $(C, Q)$ a $C_{ab}$**-curve**.

For computations in $C_{ab}$ curves, $C_{ab}$**-order** $>_{C_{ab}}$, which is a monomial order, plays a fundamental role.

**Definition 5.32** ($C_{ab}$**- order**) For $\alpha = (\alpha_1, \alpha_2)$, and $\beta = (\beta_1, \beta_2)$, let $\alpha >_{C_{ab}} \beta$ if one of the following i.e., either (i) or (ii) holds:

(i) $a\alpha_1 + b\alpha_2 > a\beta_1 + b\beta_2$
(ii) $a\alpha_1 + b\alpha_2 = a\beta_1 + b\beta_2, \alpha_1 < \beta_1$.

Note that monomials $X^\alpha Y^\beta$ are holomorphic away from $Q$, supposed to be functions on $C_{ab}$ curve. In $C_{ab}$ order, monomials $X^\alpha Y^\beta$, being supposed to be functions on $C_{ab}$ curve, are put in order by the pole number

$$-v_Q(X^\alpha Y^\beta) = a\alpha + b\beta$$

at $Q$, and when two monomials have the same pole number at $Q$, the monomials with larger degree in $X$ should be smaller.

Now we are ready to characterize the defining equation of $C_{ab}$ curves.

**Theorem 5.33** *Let $\bar{K}$ be the algebraic closure of $K$, $C \subset \bar{K}^2$ be possibly reducible affine algebraic set defined over $K$, $x$,$y$ the coordinates of the affine plane $\bar{K}^2$. The following two conditions are equivalent*

(a) *$C$ is an absolutely irreducible algebraic curve with exactly one $K$-rational place $Q$ at infinity, and the pole divisors of $x$ and $y$ are $aQ$ and $bQ$ respectively.*

(b) *$C$ is defined by a bivariate polynomial of the form*

$$F(X,Y) = \alpha_{b,0}X^b + \alpha_{0,a}Y^a + \sum_{ia+jb<ab} \alpha_{i,j}X^iY^j, \tag{5.9}$$

*where $\alpha_{i,j} \in K$ for all $i,j$ and $\alpha_{b,0}$,$\alpha_{0,a}$ are nonzero.*

**Proof** (a $\Rightarrow$ b) Let $X, Y$ be variables over $K$, $F(X,Y) \in K[X,Y]$ the defining equation of $C$, and $x, y \in K[X,Y]/F(X,Y)$ be the images of $X, Y$ respectively.

Consider the minimal polynomial $G(X,Y)$ of $y$ over the subfield $K(x)$. Since $[K(x,y) : K(x)] = \deg(x)_0 = a$ [Theorem 5.23 ]. On the other hand, $[K(x,y) : K(x)] = \deg G(x,Y)$, it follows that the degree of $G(x,Y)$ is $a$. The integral closure of $K[x]$ in $K(x,y)$ is $L(\infty Q)$ [Theorem 5.29], and $y \in L(\infty Q)$. Thus $y$ is integral over $K[x]$, and $G(x,Y) \in K[x,y]$. Write $G(X,Y)$ as

$$\sum_{X,Y} \beta_{i,j} X^iY^j.$$

If $v_Q(x^iy^j) = -ab$ and both $i$ and $j$ are nonnegative, then $(i,j)$ is either $(b,0)$ or$(0,a)$. By the strict triangle inequality of a discrete valuation [Definition 5.6], for every term $\beta_{i,j}x^iy^j$ in $G(x,y)$, we have $v_Q(x^iy^j) \geq -ab$, and the coefficient $\beta_{b,0}$ is nonzero. Therefore if $\beta_{i,j}x^iy^j$ is a term in $G(x,y)$, then the exponent $(i,j)$ is either $(b,0)$ or $(0,a)$, or $ai + bj < ab$.

Finally we have to show that $F(X,Y)$ is a constant multiple of $G(X,Y)$, and it is enough to show that $G(X,Y)$ generates the kernel of the canonical homomorphism $\phi : K[X,Y] \to K[x,y]$. If $H(X,Y) \in ker\phi \backslash 0$, then the degree of $H(X,Y)$ in $Y$ is at least $a$, because $G(x,Y)$ is the minimal polynomial of $y$ over $K(x)$. Thus $\{G(X,Y)\}$ is a Gröbner basis of $ker\phi$ with respect to the $C_{ab}$-order $>_{C_{ab}}$ (this implies that $Y >_{C_{ab}} X$, and Gröbner basis generates the ideal [Cox-Little-O'Shea [21], Corollary 6, Section 2.5].

(b $\Rightarrow$ a) Let $F(X,Y)$ be the polynomial of (5.9), and $x, y$ the elements in $K[X,Y]/F(X,Y)$ represented by $X, Y$ respectively. By the theory of Gröbner bases [Cox-Little-O'Shea [21], Proposition 4, Section 5.3], we see that

$$\{x^iy^j | 0 \leq j \leq a-1\} \tag{5.10}$$

is a basis of $K[x,y]$ as a $K$-linear space, where $\{F(X,Y)\}$ is viewed as a Gröbner basis with respect to the monomial order $Y >_{C_{ab}} X$.

Any element $f$ in $K[x,y]$ can be written uniquely as a polynomial

$$\sum_{i,j} \beta_{i,j} x^iy^j, \tag{5.11}$$

where each polynomial $x^iy^j$ belong to the basis (5.10) and $\beta_{i,j} \in K$. We define a function $o$ from $K[x,y] \to \mathbb{Z} \cup -\infty$ to be

$$\begin{aligned} o(0) &= -\infty \\ o(f) &= \max\{ak + bl | x^ky^l \text{ is a monomial of } f \text{ written as (5.11)}\}, \end{aligned}$$

where $f$ is nonzero. Then for $f, g \in K[x,y]$, $o(f) = -\infty \text{iff} f = 0$, and $o(fg) = o(f) + o(g)$, where the sum of $-\infty$ and an integer is $-\infty$.

Now we can prove the absolute irreducibility of the polynomial (5.9). The following discussion is based on (Hoholdt-Lint-Pellikaan [44], Proposition 12). Suppose that $fg = 0$ for $f, g \in K[x,y]$. Then $o(fg) = -\infty$, which implies $o(f) = -\infty$

or $o(g) = -\infty$. Thus $K[x, y]$ is an integral domain. This argument is valid if $K$ is replaced by its algebraic closure. So the polynomial (5.9) is absolutely irreducible.

Next we will show that $C$ has exactly one place at infinity. Note that $K[x, y]/K$ is an algebraic function field with the full constant field $K$. We define function $v$ from $K(x, y) \to \mathbb{Z} \cup \{\infty\}$ such that for nonzero $f/g \in K(x, y), v(f/g) = o(g) - o(f)$ and $v(0) = \infty$. Then $v$ satisfies the axiom of the discrete valuation [Definition 5.6]. Let $Q$ be the place of $K(x, y)/k$ corresponding to $v$, and $P_\infty$ be the pole of $x$ in the rational function field $K(x)$. Then $Q$ lies over $P_\infty$ and the ramification index of $Q$ over $P_\infty$ is $a$, which equals to the extension degree $[K(x, y) : K(x)]$. Thus $P_\infty$ is totally ramified, $Q$ is $K$-rational, and the pole divisor of $x$ in $K(x, y)$ is $aQ$. A similar argument shows that the pole divisor of $y$ is $bQ$. Since $K[x, y] \subseteq L(\infty Q)$, the set of places at infinity is $\{Q\}$. □

Following Definition 5.31 we give the next equivalent definition.

**Definition 5.34** *A plane curve defined by a polynomial of the form (5.9) is said to be a $C_{ab}$ curve.*

## 5.4 Addition algorithm for Jacobian group in divisor representation

Volcheck [85], has given an algorithm for addition in the Jacobian of a general curve over a finite field. However, his algorithm needs factorization of polynomials over finite fields, so its efficiency is not enough for use in cryptosystems. Moreover, the algorithm in (Volcheck [85]) needs many divisions of labor, so its space complexity is too high for real use.

We will construct much more efficient algorithm for addition in the Jacobian of $C_{ab}$ curve in this and the next section. Our algorithm has the complexity $O(g^3 \log(q)^2)$ for $C_{ab}$ curve of genus $g$ defined over $GF(q)$, and it could be tolerable for use in real cryptosystems.

Take a $C_{ab}$ curve $C$ defined over a finite field $K$. Recall $\mathfrak{D}_K, \mathfrak{D}_K^0$, and $\mathbb{P}_K$ denotes the divisor group of $C$ over $K$, its subgroup of divisors of degree zero, and the principal divisor group of $C$ over $K$, respectively. Then the Jacobian group $\mathbb{J}_K$ of $C$ over $K$ will be $\mathfrak{D}_K^0/\mathbb{P}_K$. When an element $j$ in the Jacobian $\mathbb{J}_K(C)$ has a representative $D$ in $\mathfrak{D}_K^0$, we set $j = [D]$;

$$\begin{array}{rcl} \mathbb{J}_K(C) & \simeq & \mathfrak{D}_K^0/\mathbb{P}_K, \\ j & \to & [D]. \end{array}$$

**Definition 5.35** A divisor $D$ of degree zero is called "semi-normal" when we have $D = E - nP_\infty$ with a positive divisor $E$ prime to $P_\infty$ and with a positive integer $n$ between 0 and $g$.

**Lemma 5.36** *For any element $j$ in the Jacobian $\mathbb{J}_K(C)$, there is a semi-normal divisor $D$ such that $j = [D]$.*

**Proof** There is a divisor $D$ of degree zero such that $j = [D]$. Because we have $\dim \mathcal{L}(D+gP_\infty) \geq g+1-g = 1$ from Riemann's theorem, there is a nonzero function $f \in \mathcal{L}(D + gP_\infty)$ such that $D + gP_\infty + (f) \geq 0$. Then, setting $E = D + gP_\infty + (f)$, we have $j = [E - gP_\infty]$. □

By Lemma 5.36, we can express any element $j$ in the Jacobian $\mathbb{J}_K(C)$ by a semi-normal divisor $D$. However, the expression is ambiguous, because there are

several semi-normal divisor $D_i$'s such that $j = [D_i]$. In order to choose the unique semi-normal divisor, we use the following algorithm.

### Algorithm 1

*Input:* *a divisor $D = E - nP_\infty$ of degree zero, where $E$ is a positive divisor prime to $P_\infty$*

*Output:* *a semi-normal divisor $G$ equivalent to $-D$ 1° Find a nonzero function $f \in L(\infty P_\infty)$ satisfying $(f)_0 \geq E$ with the least pole number $-v_{P_\infty}(f)$. 2° $G \leftarrow -D + (f)$.*

Algorithm 1 chooses the unique divisor associated to a given element $j \in \mathbb{J}_K(C)$. That is,

**Proposition 5.37** *For any two equivalent semi-normal divisors $D_1$ and $D_2$, the outputs of Algorithm 1 for $D_1$ and $D_2$ are the same as divisors.*

**Proof** Let $D_1 = E_1 - nP_\infty$ and $D_2 = E_2 - nP_\infty$ be equivalent semi-normal divisors. Then there is a nonzero function $\lambda$ such that

$$E_1 - n_1 P_\infty = E_2 - n_2 P_\infty + (\lambda) .$$

Take a function $f_1$ for $D_1$ just as in Algorithm 1 such that

$$\mathrm{Supp}\left((f_1)_\infty\right) \subseteq \{P_\infty\} , \quad (f_1)_0 \geq E_1 . \tag{5.12}$$

(Here, Supp($D$) denotes the support of a divisor $D$.) Then, putting the pole number of $f_1$ at $P_\infty$ as $k_1$, we have

$$\begin{aligned}(f_1\lambda^{-1}) &= (f_1) - (\lambda)\\ &= (f_1)_0 - E_1 + E_2 + (n_1 - k_1 - n_2)P_\infty .\end{aligned}$$

Here, because $(f_1)_0 - E_1 + E_2 \geq E_2$, setting $f_2 = f_1\lambda^{-1}$, $f_2$ we see that

$$\mathrm{Supp}\left((f_2)_\infty\right) \subseteq \{P_\infty\} , \quad (f_2)_0 \geq E_2 . \tag{5.13}$$

Because $\lambda$ is independent on the choice of $f_1$ and $f_2$, the correspondence $f_1 \mapsto f_2 = f_1\lambda^{-1}$ assigns $f_1$ with the least pole order at $P_\infty$ to $f_2$ with the least pole order at $P_\infty$. Thus the relation

$$\begin{aligned}-E_2 + n_2 P_\infty + (f_2) &= -E_2 + n_2 P_\infty + (f_1) - E_1 + E_2 + (n_1 - n_2)P_\infty\\ &= -E_1 + n_1 P_\infty + (f_1) .\end{aligned}$$

shows that the output of Algorithm 1 for $D_1$ is the same as the output for $D_2$ as divisors. □

**Definition 5.38** We call semi-normal divisors obtained as outputs of Algorithm 1 "normal divisors."

Algorithm 1 outputs a normal divisor which is equivalent to $-1$ times the input semi-normal divisor. So, by applying Algorithm 1 twice repeatedly to any semi-normal divisor, we get the equivalent normal divisor. From Lemma 5.36 and Proposition 5.37, we get

**Theorem 5.39** *Any element $j$ in Jacobian $\mathbb{J}_K(C)$ is expressed uniquely by a normal divisor.*

From Theorem 5.39, we get the following algorithm for addition in the Jacobian:

**Algorithm 2**

*Input: semi-normal divisors* $D_1 = E_1 - n_1 P_\infty$ *and* $D_2 = E_2 - n_2 P_\infty$
*Output: a normal divisor* $D_3 = E_3 - n_3 P_\infty$ *which is equivalent to* $D_1 + D_2$ 1° *By applying Algorithm 1 to* $D_1 + D_2 = (E_1 + E_2) - (n_1 + n_2)P_\infty$, *get a normal divisor* $D' = E' - n'P_\infty$. 2° *By applying Algorithm 1 to a normal divisor* $D' = E' - n'P_\infty$, *get a normal divisor* $D_3 = E_3 - n_3 P_\infty$ *and output* $D_3$.

To perform Algorithms 1 and 2 on computers, we need to encode divisors in some way. The most straightforward way is to encode divisors as point sets with multiplicities as in (Volcheck [85]). But to encode divisors involved in Algorithms 1 and 2 as point sets, we need to factor polynomials of degree $3g$ into irreducible components and it hurts the efficiency of the algorithms.

In $C_{ab}$ curves, Jacobian group is naturally isomorphic to the ideal class group of the coordinate ring. In the next section, appealing to this fact, we realize Algorithms 1 and 2 by ideal computations in the coordinate ring.

## 5.5 Addition algorithm for Jacobian group in ideal representation

Using the natural isomorphism between the Jacobian of $C_{ab}$ curve and the ideal class group of its coordinate ring, we realize Algorithm 1 and 2 by ideal computations.

**5.5.1 Construction of the addition algorithm.** Let $C$ be a $C_{ab}$ curve defined over a finite field $K$ with a defining equation

$$F(X,Y) = \sum_{0 \le i \le b,\ 0 \le j \le a,\ ai+bj \le ab} \alpha_{i,j} X^i Y^j = 0 \ .$$

Because $C_{ab}$ curve $C$ is nonsingular on affine plane, its coordinate ring $K[x,y] = K[X,Y]/(F(X,Y))$ is a Dedekind domain.

In general, for an algebraic curve $C$ of which coordinate ring $A_K$ is a Dedekind domain, its Jacobian group $\mathbb{J}_K(C)$ is naturally isomorphic to the ideal class group $H(A_K)$ of its coordinate ring (Hartshorne [42]). For a $C_{ab}$ curve $C$, the isomorphism $\Phi$ is given by

$$\Phi : \begin{array}{ccc} \mathbb{J}_K(C) & \tilde{\rightarrow} & H(A_K) \\ \Sigma_{P \neq P_\infty} n_P P & \mapsto & \prod I_p^{n_p} \end{array}$$

where, $I_p$ is the maximal ideal at $P$. We call ideals, corresponding to normal divisors by the isomorphism $\Phi$, normal ideals. (Note any ideal of $A_K$ corresponds to semi-normal divisors.) Recall $C_{ab}$ order is based on the pole order at $P_\infty$ of monomials as functions on $C_{ab}$ curve. Applying the isomorphism $\Phi$ to the Algorithms 1 and 2, we get the following Algorithms 3 and 4 for addition in Jacobian group $\mathbb{J}_K(C)$.

**Algorithm 3**

*Input: an ideal* $I$ *of the coordinate ring* $A$
*Output: a normal ideal* $J$ *equivalent to the inverse ideal of* $I$
1° *Find the minimum polynomialf* $\neq 0$ *w.r.t.* $C_{ab}$ *order, belonging to the ideal* $I$.
2° *Find an ideal* $J$ *satisfying* $(f) = I \cdot J$.

**Algorithm 4**

*Input: ideals* $I_1$ *and* $I_2$ *of the coordinate ring* $A$
*Output: a normal ideal* $I_3$ *equivalent to the ideal product* $I_1 \cdots I_2$

1° *By applying Algorithm 3 to the ideal product* $I_1 \cdot I_2$*, get a normal ideal* $J$.
2° *By applying Algorithm 3 to the ideal* $J$*, get a normal ideal* $I_3$.

In Algorithm 4, we must call Algorithm 3 twice. Now, we are merging two calls of Algorithm 3 in Algorithm 4. Suppose we get a polynomial $f$ and the ideal $J$ in the first call of Algorithm 3. Then, they must satisfy

$$(f) = I_1 \cdot I_2 \cdot J.$$

Similarly, suppose we get a polynomial $g$ and the ideal $I_3$ in the second call of Algorithm 3. Then,

$$(g) = J \cdot I_3.$$

Using these relations, we have

$$I_1 \cdot I_2 \cdot (g) = I_1 \cdot I_2 \cdot J \cdot I_3 \tag{5.14}$$

$$= (f) \cdot I_3 \ . \tag{5.15}$$

So, the output $I_3$ of Algorithm 4 should satisfy

$$I_3 = g/f \cdot I_1 \cdot I_2 \ .$$

Thus, we get the following Algorithm 5 for addition in the Jacobian group of $C_{ab}$ curve.

**Algorithm 5 (Addition to the Jacobian)**

*Suppose* $C_{ab}$ *curve is given by* $F(X,Y) = 0$.
*Input: Ideals* $I_1$ *and* $I_2$ *of* $K[X,Y]/(F(X,Y))$
*Output: a normal ideal* $I_3$ *of* $K[X,Y]/(F(X,Y))$

1° $J \to I_1 \cdot I_2$.
2° $f \to$ *the minimum polynomial* $f(\neq 0) \in J$. *w.r.t.* $C_{ab}$ *order*
3° $g \to$ *the minimum polynomial* $g(\neq 0)$ *w.r.t.* $C_{ab}$ *order, satisfying* $g \cdot J \subseteq \langle f, F \rangle$.
4° $I_3 \to g/f \cdot J$.

**5.5.2 Gröbner bases with respect to $C_{ab}$ order.** We express ideals in inputs or outputs for Algorithm 5 by reduced Gröbner bases w.r.t. $C_{ab}$ order. For reduced Gröbner bases w.r.t. $C_{ab}$ order, the following proposition is important.

**Proposition 5.40** *For any semi-normal divisor* $E - nP_\infty$*, and the corresponding ideal* $I$ *by the isomorphism* $\Phi$*, we have*

$$\deg(E) = \delta(I) \ .$$

*Here,* $\delta(I)$ *denotes the number of elements in the* $\Delta$*-set of ideal* $I$.

**Proof** Because two sides of the equation to be proved are unchanged under base field extensions, we can assume the definition field $K$ is algebraically closed. So, for $E = \Sigma n_P P$, we have $I = \Phi(E) = \Pi I_P^{n_P}$, where $I_P$ is the maximal ideal $P$. By Cox-Little-O'Shea [21] Chap 5. Sec 3. Prop. 4, $\delta(I) = \dim_K A/I$. On the other hand, $A/I = \Sigma_P A/I_P^{n_P}$ and as the $C_{ab}$ curve $C$ is nonsingular on the affine plane, we have $\dim_K A/I_P^{n_P} = n_P$. Therefore we obtain that $\dim A/I = \Sigma_P n_P = \deg(E)$. □

Using Proposition 5.40, we can see the form of Gröbner bases of various ideals w.r.t. $C_{ab}$ order. Some of the examples are the following:

**Example 5.41 Elliptic curve** *For an elliptic curve, an ideal corresponding to a general element has the following form of reduced Gröbner basis:*

$$\{X + a_0, Y + b_0\}.$$

Here, $a_0, b_0 \in K$.

**Verification.** As the genus of an elliptic curve is $g = 1$, a general element $j = [P - P_\infty]$ in Jacobian $\mathbb{J}_K(C)$ corresponds to a single point $\{P\}$ on the curve. With respect to $C_{23}$ order, the second monomial is $X$, the third is $Y$. So, the ideal $I$ should contain two polynomials of the forms $X + \cdots, Y + \cdots$ (Here, $\cdots$ denotes the lower terms). As we can see easily, $\delta(X + \cdots, Y + \cdots) = 1$. However, from Proposition 5.40, we see $\delta(I) = 1$. So, (5.8) in section 2.6 shows $\{X + a_0, Y + b_0\}$ is in fact a Gröbner basis of $I$. □

**Example 5.42 Hyperelliptic curve** *For an hyperelliptic curve, an ideal corresponding to a general element in Jacobian group $\mathbb{J}_K(C)$ for $C_{2\{2g+1\}}$ curve has the following form of reduced Gröbner basis:*

$$\left\{X^g + a_{g-1}X^{g-1} + \cdots + a_0, Y + b_{g-1}X^{g-1} + \cdots + b_0\right\}$$

where, $a_i, b_i \in K$

**Verification.** In this case the genus of $C$ is $g(C) = (2-1)(2g+1-1)/2 = g$, a general element $j = [P_1 + \cdots + P_g - gP_\infty]$ in Jacobian $\mathbb{J}_K(C)$ corresponds to $g$ points $\{P_1, \cdots, P_g\}$ on the curve $C$. With respect to $C_{2\{2g+1\}}$ order, the monomials stand in a row

$$1, X, X^2, \cdots, X^g, Y, X^{g+1}, XY, X^{g+2}, X^2y, X^{g+3}, \cdots, X^{g+d}, X^dY, \cdots$$

in ascending order. One can easily check that

$$\delta(X^g + \cdots, Y + \cdots) = g$$

(Here, $\cdots$ denotes the lower terms).

On the other hand, from Proposition 5.40, we see $\delta(I) = g$ Thus, (5.8) in section 2.6 shows

$$\left\{X^g + a_{g-1}X^{g-1} + \cdots + a_0, Y + b_{g-1}X^{g-1} + \cdots + b_0\right\}$$

is in fact a Gröbner basis of $I$. □

**Example 5.43 Superelliptic curve of type $C_{34}$** *An ideal corresponding to a general element in Jacobian group $\mathbb{J}_K(C)$ for $C_{34}$ curve $C$ has the following form of reduced Gröbner basis:*

$$\left\{a_0 + a_1X + a_2Y + X^2,\ b_0 + b_1X + b_2Y + XY,\ c_0 + c_1X + c_2Y + Y^2\right\}\ .$$

*Here, $a_i, b_i, c_i \in K$.*

**Verification.** As the genus of $C$ is $g(C) = (3-1)(4-1)/2 = 3$, a general element $j = [P_1 + P_2 + P_3 - 3P_\infty]$ in Jacobian $\mathbb{J}_K(C)$ corresponds to three points $\{P_1, P_2, P_3\}$ on the curve $C$. With respect to $C_{34}$ order, the fourth monomial is $X^2$, the fifth is $XY$, and the sixth is $Y^2$. So, the ideal $I$ should contain three polynomials of the forms $X^2 + \cdots, XY + \cdots, Y^2 + \cdots$ (Here, $\cdots$ denotes the lower terms). As we can see easily, $\delta(X^2 + \cdots, XY + \cdots, Y^2 + \cdots) = 3$. However, from Proposition 5.40, we see $\delta(I) = 3$. So, (5.8) in section 2.6 shows $\left\{X^2 + \cdots, XY + \cdots, Y^2 + \cdots\right\}$ is in fact a Gröbner basis of $I$. □

**Example 5.44 Superelliptic curve of type $C_{35}$** *For an superelliptic curve of type $C_{35}$, an ideal corresponding to a general element in Jacobian group $\mathbb{J}_K(C)$ for has the following form of reduced Gröbner basis:*

$$\{XY+a_0x^2+a_1Y+a_2X+a_3, X^3+b_0X^2+b_1Y+b_2X+b_3, Y^2+c_0X^2+c_1Y+c_2X+c_3\}$$

where, $a_i, b_i, c_i \in K$

**Verification.** In this case the genus of $C$ is $g(C) = (3-1)(5-1)/2 = 4$, a general element $j = [P_1 + \cdots + P_4 - 4P_\infty]$ in Jacobian $\mathbb{J}_K(C)$ corresponds to four points $\{P_1, \cdots, P_4\}$ on the curve $C$. With respect to $C_{35}$ order, the monomials stand in a row

$$1, X, Y, X^2, XY, X^3, Y^2, X^2Y, X^4, XY^2, X^3Y, X^5, X^2Y^2, X^4Y, \cdots$$

in ascending order. One can easily check that

$$\delta(XY + \cdots, X^3 + \cdots, Y^2 + \cdots) = 4$$

(Here, $\cdots$ denotes the lower terms).

On the other hand, from Proposition 5.40, we see $\delta(I) = 4$. Thus, (5.8) in Section 2.6 shows

$$\{XY+a_0X^2+a_1Y+a_2X+a_3, X^3+b_0X^2+b_1Y+b_2X+b_3, Y^2+c_0X^2+c_1Y+c_2X+c_3\}$$

is in fact a Gröbner basis of $I$. □

**Example 5.45 Superelliptic curve of type $C_{56}$** *For an superelliptic curve of type $C_{56}$, an ideal corresponding to a general element in Jacobian group $\mathbb{J}_K(C)$ for has the following form of reduced Gröbner basis:*

$$\{X^4 + \cdots, X^3y + \cdots, X^2Y^2 + \cdots, XY^3 + \cdots, Y^4 + \cdots\}$$

where, $\cdots$, denotes a linear combination of lower terms, i.e.,

$$1, X, Y, X^2, XY, Y^2, X^3, X^2y, XY^2, Y^3, X^4$$

with coefficients in $K$.

**Verification.** In this case the genus of $C$ is $g(C) = (5-1)(6-1)/2 = 10$, a general element $j = [P_1 + \cdots + P_{10} - 10P_\infty]$ in Jacobian $\mathbb{J}_K(C)$ corresponds to ten points $\{P_1, \cdots, P_{10}\}$ on the curve $C$. With respect to $C_{56}$ order, the monomials stand in a row

$$1, X, Y, X^2, XY, Y^2, X^3, X^2Y, XY^2, Y^3, x^4, X^3Y, X^2Y^2, XY^3, Y^4, \cdots$$

in ascending order. One can easily check that

$$\delta(X^4 + \cdots, X^3Y + \cdots, X^2Y^2 + \cdots, XY^3 + \cdots, Y^4 + \cdots) = 10$$

Here $\cdots$, denotes the lower terms as before).

On the other hand, from Proposition 5.40, we see $\delta(I) = 10$. Thus, (5.8) in Section 2.6 shows

$$\{X^4 + \cdots, X^3Y + \cdots, X^2Y^2 + \cdots, XY^3 + \cdots, Y^4 + \cdots\}$$

is in fact a Gröbner basis of $I$. □

Although we dealt with the ideal corresponding to $g$ points on $C_{ab}$ curve in the above examples, the situation is similar for ideals corresponding to any number of points on general $C_{ab}$ curve.

**5.5.3 Details of addition algorithm.** We show an example of performing Algorithm 5 and explain details of the algorithm. For instance, we take $C_{34}$ curve $C : Y^3 + X^4 + 1 = 0$ over a prime field $GF(17)$ and compute twice of an ideal $I = \{f_1 = X^2 + 14Y + 4X + 5,\ f_2 = XY + 3Y + 4X + 9,\ f_3 = Y^2 + 9Y + 16X + 2\}$ which corresponds to an element of Jacobian $\mathbb{J}_{GF(17)}$ (**Example** 5.43). In $C_{34}$ order, monomials stand in a row

$$1, X, Y, X^2, XY, Y^2, X^3, X^2Y, XY^2, \cdots$$

in ascending order.

1° Compute the ideal product $I \cdot I$.

As $I \cdot I$ corresponds to six elements on the curve $C$ and the seventh monomial in $C_{34}$ order is $X^3$, we can see that the Gröbner basis of $I \cdot I$ has the form of $\{X^3 + \cdots, X^2Y + \cdots, XY^2 + \cdots\}$, just as in the verification of (5.43). From now on, $\bar{f}^G$ denotes the remainder of $f$ divided by a polynomial set $G$.

Now, we begin computations:

$$\begin{aligned}
g_1 &\to \overline{f_1^2}^{\{F\}} = X^4 + \cdots \\
g_2 &\to \overline{f_1 \cdot f_2}^{\{g_1, F\}} = X^3Y + \cdots \\
g_3 &\leftarrow \overline{f_2^2}^{\{g_2, g_1, F\}} = X^2Y^2 + \cdots \\
g_4 &\leftarrow \overline{f_1 \cdot f_3}^{\{g_3, g_2, g_1, F\}} = XY^2 + \cdots \\
g_5 &\leftarrow \overline{f_2 \cdot f_3}^{\{g_4, g_3, g_2, g_1, F\}} = X^2Y + \cdots \\
g_6 &\leftarrow \overline{f_3^2}^{\{g_5, g_4, g_3, g_2, g_1, F\}} = X^3 + \cdots
\end{aligned}$$

So, we have

$$J \leftarrow I \cdot I = \{g_6, g_5, g_4\}\ .$$

This is the expression of $J$ by the Gröbner basis.

2° $f \leftarrow g_6 = X^3 + 10Y^2 + 5XY + 7Y + 11X + 4$.

3° Find the minimum polynomial $h$ satisfying $h \cdot J \subset \langle f, F \rangle$.

Note $\{f, F\}$ is a Gröbner basis of the ideal $\langle f, F \rangle$ as $LT(f) = X^3$ and $LT(F) = Y^3$ are prime to each other. We compute products of $g_5$ and monomials modulo $\{f, F\}$ in ascending order:

$$\begin{aligned}
\overline{g_5}^{\{f,F\}} &= X^2Y + \cdots \\
\overline{Xg_5}^{\{f,F\}} &= XY^2 + \cdots \\
\overline{Yg_5}^{\{f,F\}} &= X^2Y^2 + \cdots
\end{aligned}$$

Now, computing $\overline{X^2g_5}^{\{f,F\}}$, we encounter $4X^2Y^2 + \cdots$ . But its leading monomial $X^2Y^2$ is the same as the one of $\overline{Yg_5}^{\{f,F\}}$. So, we have $X^2g_5 \equiv 4Yg_5 + 12XY^2 + \cdots \pmod{\{f, F\}}$. Moreover, we note $XY^2$ is the leading term of $\overline{Xg_5}^{\{f,F\}}$, and repeat similar computations as above. We get

$$X^2g_5 \equiv 4Yg_5 + 12Xg_5 + 2g_5 \pmod{\{f, F\}}\ .$$

Thus, we obtain $h \leftarrow X^2 + 13Y + 5X + 15$.

4°

$$\begin{aligned}(h/f)\cdot J &= (h/f)\cdot\{g_6, g_5, g_4\}\\ &= \{h, (hg_5)/f,\ (hg_4)/f\}\ .\end{aligned}$$

So, we compute remainder of $hg_5$ and $hg_4$ divided by $\{f, F\}$ and suppose we get $\{a_5, b_5\}$ and $\{a_4, b_4\}$, respectively. Then,

$$\begin{aligned}I_3 &\leftarrow \{h, (hg_5)/f, (hg_4)/f\}\\ &\equiv \{h, a_5, a_4\} \quad (\text{mod } \{F\})\\ &= \{X^2 + 13Y + 5X + 15, XY + 13Y + 5X + 11, Y^2 + 5Y + 12X + 6\}.\end{aligned}$$

This is the expression of $I_3$ by the Gröbner basis.

From the above argument, we obtain Algorithm 6 for addition in Jacobian, giving details of Algorithm 5. In the following, "$\{\{c_1, c_2, \ldots, c_a\}, r\} \leftarrow$ Division(g, G)" denotes that we get the quotient $\{c_1, c_2, \ldots, c_a\}$ and the remainder $r$ by dividing the polynomial $g$ by the polynomial set $G$ (see [CLO] for details). "$\{\{a_1, \ldots, a_i\}, r\} \leftarrow$ Coefficients$(\mathrm{f}, \mathrm{r}_1, \ldots, \mathrm{r}_\mathrm{i})$" denotes that we get coefficients $\{a_1, \ldots, a_i\}$ and the remainder $r$ to express $f$ as a linear combination of $r_1, \ldots, r_i$. $Mono_i$ denotes the $i$-th monomial in $C_{ab}$ order ($Mono_1 = 1,\ Mono_2 = X, \ldots$).

**Algorithm 6 (Addition algorithm for Jacobian)**

```
algorithm JacobianSum(inputs I_1, I_2, output I_3)
    I_3 ← Compose(I_1, I_2)
    f ← the minimum element of I_3
    I_3 ← Reduce(f, I_3)
    RETURN I_3

subroutine Compose(inputs I_1 = {f_1, f_2, ..., f_a}, I_2 = {g_1, g_2, ..., g_a}, output
I_3)
    I_3 ← {F}
    FOR i = 1 TO a, j = 1 TO a DO
        g ← \overline{f_i · g_j}^{I_3}
        I_3 ← {g} ∪ I_3
    IF δ(I_3) > δ(I_1) + δ(I_2) THEN I_3 ← Buchberger(δ(I_1) + δ(I_2), I_3)
    I_3 ← the set of the minimum a elements of I_3
    RETURN I_3

subroutine Reduce(inputs f, I = {f_1, f_2, ..., f_a}, output J)
    G ← { f, \overline{f · y}^{F}, ..., \overline{f · y^{a-1}}^{F} }
    LABEL(retry)
    J ← {}
    h ← (random number)·f_1+ (random number)·f_2+···+ (random number)·f_a
    g ← Divide(G, h)
    FOR i = 1 TO a
        {{c_1, c_2, ..., c_a}, r} ← Division(g · f_i, G)
        IF r ≠ 0 THEN GOTO retry
        k ← c_1 + c_2 · y + ··· + c_a · y^{a-1}
        J ← J ∪ {k}
    RETURN J

subroutine Divide(inputs G, h, output s)
```

```
    r_1 ← \overline{Mono_1 · h}^G
    s_1 ← Mono_1
    i ← 1
    WHILE r_i ≠ 0 DO
         i ← i + 1
         r_i ← \overline{Mono_i · h}^G
         {{A_1, ..., A_{i-1}}, r_i} ← Coefficients(r_i, {r_1, ..., r_{i-1}})
         s_i ← Mono_i − Σ_{j=1}^{i-1} A_j s_j
    RETURN s_i
subroutine Buchberger(inputs m, I = {f_1, ..., f_s}, output G = {g_1, ..., g_t})
    B ← {(i,j)|1 ≤ i < j ≤ s}
    G ← F
    t ← s
    WHILE B ≠ φ AND δ(G) > m DO
         Select (i,j) ∈ B
         IF LCM(LT(f_i),LT(f_j)) ≠ LT(f_i)LT(f_j) THEN
              S ← \overline{S(f_i, f_j)}^G
              IF S ≠ 0 THEN
                   t ← t + 1; f_t ← S
                   G ← G ∪ {f_t}
                   B ← B ∪ {(i,t)|1 ≤ i < t − 1}
         B ← B − {(i,j)}
    RETURN G
```

How about the time complexity of Algorithm 6? It is clear that the complexity is dominated by the subroutine Buchberger. In Algorithm 6, we always know the order of $\Delta$–set of ideals before knowing their Gröbner bases, using Proposition 8. So, subroutine Buchberger in Algorithm 6 monitors the order of the $\Delta$–set of $G$ and when the order becomes equal to the input $m$, the subroutine can finish at once. Therefore, if we make a suitable scheduling for selecting $(i, j)$ from $B$ in the subroutine Buchberger, we can see that the complexity of the subroutine Buchberger, as well as the complexity of Algorithm 6, evaluated by the number of multiplications on the field of definition, is $O(g^3)$.

However, our experimental results show, when the size of the field of definition is large enough, e.g. scores of bits, Gröbner bases are already obtained before performing subroutine Buchberger in subroutine Compose, almost always. Therefore, even without such scheduling, the complexity of Algorithm 6, evaluated by the number of multiplications on the finite field, is actually $O(g^3)$ with large fields of definition.

**Farzali Izadi**
Department of Mathematics
Azarbaijan University of Tarbiat Moallem
Tabriz, Iran
farzali.izadi@gmail.com

Fields Institute Communications
Volume **58**, 2010

CHAPTER 6

# The Zeta Functions of Two Garcia-Stichtenoth Towers

Kenneth W. Shum

## 6.1 Introduction

Based on modular curves, Tsfasman, Vlăduţ and Zink showed that there exists a sequence of algebraic curves over finite field of size $q^2$ for some prime power $q$, so that the Drinfeld-Vlăduţ bound is attained. This sequence of curves can be used to construct sequence of error-correcting codes that are better than the Gilbert-Varshamov bound asymptotically [83]. However, the construction is not explicitly described. The first explicit construction that attains the Drinfeld-Vlăduţ bound was studied by Garcia and Stichtenoth [31] in the language of function fields. Later, in [32], Garcia and Stichtenoth provided another example, which is simpler to analyze. In both constructions, the extensions of function fields are Artin-Schreier. They are also examples of $C_{ab}$ curves.

Motivated by the discrete log problem over the Jacobian of these function fields, we are interested in computing their zeta functions. Once the zeta function is known, we can readily obtain the cardinality of the Jacobian.

Background on zeta functions is contained in the next section. We will also describe how to compute zeta functions by counting rational points over extensions of constant field. The zeta functions of the first and second towers over $\mathbb{F}_4$ are computed in Section 6.3 and 6.4. In the appendix we provide the details about counting of points above a ramified place in the first tower.

## 6.2 Background on zeta functions

In this section, we review the basic properties of zeta function and discuss the computation of zeta function. References for zeta function can be found in [70, 81].

For a function field $\mathcal{F}$ defined over $\mathbb{F}_q$ with genus $g = g(\mathcal{F})$, and for $r \geq 1$ let $N_r = N_r(\mathcal{F})$ be the number of rational places over the extension field $\mathbb{F}_{q^r}$. The *zeta function* of $\mathcal{F}$ over $\mathbb{F}_q$ is defined as

$$Z(T, \mathcal{F}/\mathbb{F}_q) := \exp\Big(\sum_{r=1}^{\infty} \frac{N_r T^r}{r}\Big).$$

The zeta function is a rational function of $T$ and can be written as

$$Z(T, \mathcal{F}/\mathbb{F}_q) = \frac{L(T, \mathcal{F}/\mathbb{F}_q)}{(1-T)(1-qT)}$$

where

$$L(T, \mathcal{F}/\mathbb{F}_q) = 1 + c_1 T + c_2 T + \ldots + c_{2g} T^{2g}$$

is a polynomial of degree $2g$ with integral coefficients, called the *$L$-polynomial* of $\mathcal{F}/\mathbb{F}_q$. If the function field $\mathcal{F}$ and constant field $\mathbb{F}_q$ are clear from the context, we will simply use the notation $Z(T)$ and $L(T)$ to stand for $Z(T, \mathcal{F}/\mathbb{F}_q)$ and $L(T, \mathcal{F}/\mathbb{F}_q)$ respectively.

Let $\alpha_i$, $i = 1, 2, \ldots, 2g$, be the reciprocals of the roots of $L(T)$,

$$L(T) = \prod_{i=1}^{2g}(1 - \alpha_i T).$$

By the Hasse–Weil theorem, we have $|\alpha_i| = \sqrt{q}$ for all $i$. For $r \geq 1$, the number of rational places of $\mathcal{F}$ over the extension field $\mathbb{F}_{q^r}$ is determined by

$$N_r = q^r + 1 - \sum_{i=1}^{2g} \alpha_i^r.$$

If we know $N_r$ for $r = 1, 2, \ldots, g$, we can compute the $L$-polynomial as follow (see [81, V.1.17]). Let

$$S_r := \sum_{i=1}^{2g} \alpha_i^r = q^r + 1 - N_r$$

be the the sum of the $r$th power of $\alpha_i$. We can solve for the coefficients $c_1, \ldots, c_g$ recursively using Newton's identity,

$$-c_1 = S_1 \tag{6.1}$$

$$-2c_2 = c_1 S_1 + S_2 \tag{6.2}$$

$$-3c_3 = c_2 S_1 + c_1 S_2 + S_3 \tag{6.3}$$

$$\vdots$$

$$-gc_g = c_{g-1} S_1 + c_{g-2} S_2 + \ldots + S_g \tag{6.4}$$

Then, from the functional equation

$$L\Big(\frac{1}{qT}\Big) = \frac{1}{q^g T^{2g}} L(T),$$

the coefficients $c_{g+1}, \ldots, c_{2g}$ can be obtained by

$$c_{2g-i} = q^{g-i} c_i \tag{6.5}$$

for $i = 0, 1, \ldots, g-1$. (Here we set $c_0 = 1$.)

We summarize the procedure for the computation of $L$-function as follow.

1. Count the number of rational places of $\mathcal{F}$ over $\mathbb{F}_{q^r}$ for $r = 1, 2, \ldots, g$.
2. Compute $c_1, c_2 \ldots, c_g$ by (6.1) to (6.4).
3. Compute $c_{g+1}, c_{g+2}, \ldots, c_{2g}$ by (6.5).

The number of places of extension field $\mathbb{F}_{q^r}$ increases rapidly as $r$ increases. It will take a long time to count the rational places over $\mathbb{F}_{q^r}$ when $r$ is large. In the following, we present two computational tricks that expedite the counting process.

We call the first trick *constant field reduction*. Let $p$ be the characteristic of $\mathbb{F}_q$. Suppose that the defining equations of the function field have coefficients in $\mathbb{F}_p$. We can consider $\mathcal{F}/\mathbb{F}_p$ as the function field defined by the same set of equations as in $\mathcal{F}/\mathbb{F}_q$ but over $\mathbb{F}_p$ instead. The genus is invariant under such constant field reduction, i.e., $g(\mathcal{F}/\mathbb{F}_q) = g(\mathcal{F}/\mathbb{F}_p)$. We count the number of rational places of $\mathcal{F}/\mathbb{F}_p$ over the extension field $\mathbb{F}_{p^r}$, for $r = 1, 2, \ldots, g$. The number of places over

$\mathbb{F}_{p^r}$ will be less than that over $\mathbb{F}_{q^r}$. After obtaining the $L$-polynomial $L(T, \mathcal{F}/\mathbb{F}_p)$ using the procedure described above, we then recover the $L$-polynomial of $\mathcal{F}/\mathbb{F}_q$ by

$$L(T^n, \mathcal{F}/\mathbb{F}_q) = \prod_{\xi} L(\xi T, \mathcal{F}/\mathbb{F}_p) \tag{6.6}$$

with the product taken over all complex $n$th roots of unity.

The second trick exploits the analogue of Artin's holomorphy conjecture in the function field case, which states that $L(T, \mathcal{E}/\mathbb{F}_q)$ is divisible by $L(T, \mathcal{F}/\mathbb{F}_q)$, where $\mathcal{E}/\mathcal{F}$ is a finite extension of function fields. In other words,

$$\frac{L(T, \mathcal{E}/\mathbb{F}_q)}{L(T, \mathcal{F}/\mathbb{F}_q)}$$

is a polynomial in $T$. This conjecture was proved by Kleiman [49, Prop. 1.2.4]. See also [7, Prop. 5] for an alternate proof.

Let $\alpha_i$, $i = 1, 2, \ldots, 2g(\mathcal{E})$, be the reciprocal roots of $L(T, \mathcal{E})$, and $\beta_j$, $j = 1, 2, \ldots, 2g(\mathcal{F})$, be the reciprocal roots of $L(T, \mathcal{F})$. Artin's conjecture says that, after appropriate re-labeling, we have $\alpha_i = \beta_i$ for $i = 1, 2, \ldots, 2g(\mathcal{F})$. Suppose that we have already obtained the $L$-polynomial of $\mathcal{F}$. We then know the first $2g(\mathcal{F})$ reciprocal roots of $L(T, \mathcal{E}/\mathbb{F}_q)$. It remains to find the remaining $\beta_i$ for

$$i = 2g(\mathcal{F}) + 1, \ldots, 2g(\mathcal{E}) - 1, 2g(\mathcal{E}).$$

This can be done via the power sums,

$$\begin{aligned}\tilde{S}_r &:= \sum_{i=2g(\mathcal{F})+1}^{2g(\mathcal{E})} \beta_i^r \\ &= q^r + 1 - N_r(\mathcal{E}) - \sum_{i=1}^{2g(\mathcal{F})} \alpha_i^r \\ &= N_r(\mathcal{F}) - N_r(\mathcal{E}) \end{aligned} \tag{6.7}$$

for $r = 1, 2, \ldots, g(\mathcal{E}) - g(\mathcal{F})$. The polynomial

$$\prod_{i=2g(\mathcal{F})+1}^{2g(\mathcal{E})} (1 - \beta_i T) = \frac{L(T, \mathcal{E}/\mathbb{F}_q)}{L(T, \mathcal{F}/\mathbb{F}_q)}$$

can be obtained by Newton's identity.

## 6.3 The first Garcia-Stichtenoth tower

The first *Garcia-Stichtenoth tower* (GS1) of function fields over $\mathbb{F}_{q^2}$

$$\mathcal{E}_1 \subset \mathcal{E}_2 \subset \mathcal{E}_3 \subset \ldots$$

is introduced in [31].

$\mathcal{E}_1 := \mathbb{F}_{q^2}(x_1)$ is the rational function field over $\mathbb{F}_{q^2}$. For $n \geq 1$, $\mathcal{E}_{n+1}$ is defined recursively by $\mathcal{E}_{n+1} := \mathcal{E}_n(z_{n+1})$, where

$$z_{n+1}^q + z_{n+1} = x_n^{q+1}, \tag{6.8}$$

with $x_n = z_n/x_{n-1}$ for $n \geq 2$.

It is shown in [31] that

1. $\mathbb{F}_{q^2}$ is the full constant field of $\mathcal{E}_n$, for $n \geq 1$.

2. The genus of $\mathcal{E}_n$ is

$$g(\mathcal{E}_n) = \begin{cases} q^n + q^{n-1} - \frac{1}{2}q^{(n/2)+1} - \frac{3}{2}q^{n/2} - q^{(n/2)-1} + 1 & \text{for even } n, \\ q^n + q^{n-1} - q^{(n+1)/2} - 2q^{(n-1)/2} + 1 & \text{for odd } n. \end{cases}$$

The ramified places in the tower lie above two places in the first level: the infinite place and the zero of $x_1$, denoted by $P_\infty$ and $P_0$ respectively. The infinite place is totally ramified throughout the tower, meaning that there is exactly one point in each level lying above $P_\infty$. The ramification behavior of $P_0$ is more complex. We will count the places lying above $P_0$ and calculate their degrees in the appendix.

The defining equation (6.8) for the second level of GS1 can be re-written in terms of the trace and norm from $\mathbb{F}_{q^2}$ to $\mathbb{F}_q$,

$$\mathsf{T}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(z_2) = \mathsf{N}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(x_1).$$

For each $x_1 \in \mathbb{F}_{q^2}^*$, the right hand side of the above equation is an element in $\mathbb{F}_q^*$. Solving for $z_2$, we see that there are exactly $q$ solutions for $z_2$ in $\mathbb{F}_{q^2}^*$.

The system of equations for $\mathcal{E}_{n+1}$ is

$$\begin{aligned} \mathsf{T}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(z_{k+1}) &= \mathsf{N}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(x_k), \\ x_k &= z_k/x_{k-1}, \quad k = 2, \ldots, n. \end{aligned}$$

Repeating the argument in the last paragraph, we conclude that there are $q^n$ solutions for each $x_1 \in \mathbb{F}_{q^2}^*$. (There is no division by zero because $x_1 \neq 0$.) For $\beta_1 \in \mathbb{F}_{q^2}^*$, let $P_{\beta_1}$ be the zero of $x_1 - \beta_1$ in $\mathcal{E}_1$. There are thus $q^n$ rational places lying above the place $P_{\beta_1}$, and $P_{\beta_1}$ *splits completely* in the tower for $\beta_1 \in \mathbb{F}_{q^2}^*$.

By counting the rational places above the infinity place and $P_{\beta_1}$ for $\beta_1 \in \mathbb{F}_{q^2}^*$, the number of rational places in $\mathcal{E}_n$ is larger than or equal to $1 + (q^2 - 1)q^{n-1}$. By taking the limit

$$N(\mathcal{E}_n)/g(\mathcal{E}_n) \to \frac{(q^2-1)q^{n-1}}{q^n + q^{n-1}} = q - 1$$

as $n$ approaches infinity, we see that this tower of function fields attains the Drinfeld-Vlăduţ bound [81, V.3.6].

The zeta functions for the first few levels in GS1 are obtained from counting places in $\mathbb{F}_{2^r}$ for $r = 1, 2, \ldots, 20$. Apart from the places above $P_\infty$ and $P_\beta$, for $\beta \in \mathbb{F}_{q^2}^*$, we have to (i) analyze the places above $P_0$ and (ii) count all other unramified places of degree two or higher over $\mathbb{F}_{q^2}$. (i) is done in the appendix in this chapter, and (ii) is performed by running a computer program. The zeta functions for the first 5 levels of GS1 with $q^2 = 4$ are:

$$Z(T, \mathcal{E}_1/\mathbb{F}_4) = \frac{1}{(1-T)(1-4T)} \quad \text{(genus} = 0)$$
$$Z(T, \mathcal{E}_2/\mathbb{F}_4) = \frac{(1+2T)^2}{(1-T)(1-4T)} \quad \text{(genus} = 1)$$
$$Z(T, \mathcal{E}_3/\mathbb{F}_4) = \frac{\alpha(T)^4\beta(T)^2\gamma(T)}{(1-T)(1-4T)} \quad \text{(genus} = 5)$$
$$Z(T, \mathcal{E}_4/\mathbb{F}_4) = \frac{\alpha(T)^8\beta(T)^6\gamma(T)^3}{(1-T)(1-4T)} \quad \text{(genus} = 13)$$
$$Z(T, \mathcal{E}_5/\mathbb{F}_4) = \frac{\alpha(T)^{12}\beta(T)^{10}\gamma(T)^7\delta(T)^4\epsilon(T)^2\eta(T)^2}{(1-T)(1-4T)} \quad \text{(genus} = 33)$$

where

$$\alpha(T) = 1 + 2T$$
$$\beta(T) = 1 + 4T^2$$
$$\gamma(T) = 1 + 3T + 4T^2$$
$$\delta(T) = 1 + 2T + 4T^2$$
$$\epsilon(T) = 1 - T + 4T^2$$
$$\eta(T) = 1 - 4T^2 + 16T^4.$$

## 6.4 The second Garcia-Stichtenoth tower

In the second Garcia-Stichtenoth tower (GS2) [32]

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \subset \ldots,$$

$\mathcal{H}_1 := \mathbb{F}_{q^2}(x_1)$ is the rational function field over $\mathbb{F}_{q^2}$, and for $n \geq 1$, $\mathcal{H}_{n+1}$ is defined by $\mathcal{H}_n(x_{n+1})$ with $x_{n+1}$ satisfying

$$x_{n+1}^q + x_{n+1} = \frac{x_n^q}{x_n^{q-1} + 1}.$$

This can also be written as

$$\mathsf{T}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(x_{n+1}) = \frac{1}{\mathsf{T}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(x_n^{-1})}.$$

It is shown in [32] that

1. $\mathbb{F}_{q^2}$ is the full constant field of $\mathcal{H}_n$ for $n \geq 1$.
2. The genus of $\mathcal{H}_n$ is

$$g(\mathcal{H}_n) = \begin{cases} (q^{n/2} - 1)^2 & \text{for even } n \\ (q^{(n-1)/2} - 1)(q^{(n+1)/2} - 1) & \text{for odd } n. \end{cases}$$

The ramification and splitting of places in GS2 is presented in [3]. Using the same procedure as in GS1, the zeta functions for the first 5 levels in GS2 over $\mathbb{F}_4$ are calculated.

$$Z(T,\mathcal{H}_1/\mathbb{F}_4) = \frac{1}{(1-T)(1-4T)} \quad (\text{genus} = 0)$$
$$Z(T,\mathcal{H}_2/\mathbb{F}_4) = \frac{1+3T+4T^2}{(1-T)(1-4T)} \quad (\text{genus} = 1)$$
$$Z(T,\mathcal{H}_3/\mathbb{F}_4) = \frac{(1+3T+4T^2)^3}{(1-T)(1-4T)} \quad (\text{genus} = 3)$$
$$Z(T,\mathcal{H}_4/\mathbb{F}_4) = \frac{(1+3T+4T^2)^7(1-T+4T^2)^2}{(1-T)(1-4T)} \quad (\text{genus} = 9)$$
$$Z(T,\mathcal{H}_5/\mathbb{F}_4) = \frac{a(T)^{11}b(T)^4c(T)^2d(T)^2}{(1-T)(1-4T)} \quad (\text{genus} = 21)$$

where

$$\begin{aligned} a(T) &= 1+3T+4T^2 \\ b(T) &= 1-T+4T^2 \\ c(T) &= 1+T+4T^2 \\ d(T) &= 1+2T+T^2+8T^3+16T^4. \end{aligned}$$

**Remark.** The zeta function of the sixth level in GS2 is computed in [89] by a different method.

In the rest of this section, we will illustrate the computation of $Z(T,\mathcal{H}_3/\mathbb{F}_4)$ and $Z(T,\mathcal{H}_4/\mathbb{F}_4)$.

Let $\mathcal{H}_3'/\mathbb{F}_2$ be the constant field reduction of $\mathcal{H}_3/\mathbb{F}_4$. The number of rational places of $\mathcal{H}_3'$ over $\mathbb{F}_2$, $\mathbb{F}_4$ and $\mathbb{F}_8$ are respectively

$$N_1(\mathcal{H}_3') = 4, \quad N_2(\mathcal{H}_3') = 14, \quad N_3(\mathcal{H}_3') = 4.$$

From the power sum

$$\begin{aligned} S_1 &= 2+1-N_1(\mathcal{H}_3') = -1 \\ S_2 &= 4+1-N_2(\mathcal{H}_3') = -9 \\ S_3 &= 8+1-N_3(\mathcal{H}_3') = 5 \end{aligned}$$

we obtain the coefficients

$$\begin{aligned} c_1 &= -S_1 = 1 \\ c_2 &= -\frac{1}{2}(c_1S_1+S_2) = 5 \\ c_3 &= -\frac{1}{3}(c_2S_1+c_1S_2+S_3) = 3. \end{aligned}$$

The $L$-polynomial of $\mathcal{H}_3'$ is

$$\begin{aligned} L(T,\mathcal{H}_3'/\mathbb{F}_2) &= 1+T+5T^2+3T^3+10T^4+4T^5+8T^6 \\ &= (1-T+2T^2)(1+T+2T^2)^2. \end{aligned} \tag{6.9}$$

Hence by (6.6),

$$\begin{aligned} L(T^2,\mathcal{H}_3/\mathbb{F}_4) &= L(T,\mathcal{H}_3'/\mathbb{F}_2)\cdot L(-T,\mathcal{H}_3'/\mathbb{F}_2) \\ &= (1+3T^2+4T^4)^3. \end{aligned}$$

We thus obtain

$$L(T, \mathcal{H}_3/\mathbb{F}_4) = (1 + 3T + 4T^2)^3.$$

We apply constant field reduction and consider $\mathcal{H}'_4/\mathbb{F}_2$ instead of $\mathcal{H}_4/\mathbb{F}_4$. The genus of $\mathcal{H}'_4/\mathbb{F}_2$ is equal to 9. The normal procedure for determining the zeta function of $\mathcal{H}'_4/\mathbb{F}_2$ requires counting of places over $\mathbb{F}_{2^r}$ for $r = 1, 2, \ldots, 9$. However, since we have already computed the zeta function of $\mathcal{H}'_3/\mathbb{F}_2$, we know that $L(T, \mathcal{H}'_3/\mathbb{F}_2)$ is a factor of $L(T, \mathcal{H}'_4/\mathbb{F}_2)$. It suffices to count the number of rational places over $\mathbb{F}_{2^r}$ for $r = 1, 2, \ldots, 6$:

$$\begin{aligned} N_1(\mathcal{H}'_4) &= 4, \quad N_2(\mathcal{H}'_4) = 24, \quad N_3(\mathcal{H}'_4) = 4, \\ N_4(\mathcal{H}'_4) &= 24, \quad N_5(\mathcal{H}'_4) = 44, \quad N_6(\mathcal{H}'_4) = 24. \end{aligned}$$

Before we proceed, we first determine $N_4(\mathcal{H}'_3)$, $N_5(\mathcal{H}'_3)$ and $N_6(\mathcal{H}'_3)$ from the $L$-polynomial of $\mathcal{H}'_3$. From (6.9), we let the coefficients of $L(T, \mathcal{E}'_3/\mathbb{F}_2)$ be $d_i$, for $i = 0, \ldots, 6$.

$$L(T, \mathcal{E}'_3/\mathbb{F}_2) = d_0 + d_1T + \cdots + d_6T^6.$$

Using Newton's identity, we compute the power sum $S_4$, $S_5$ and $S_6$,

$$\begin{aligned} S_4 &= -(4d_4 + d_3S_1 + d_2S_2 + d_1S_3) = 3 \\ S_5 &= -(5d_5 + d_4S_1 + d_3S_2 + \cdots + d_1S_4) = -11 \\ S_6 &= -(6d_6 + d_5S_1 + d_4S_2 + \cdots + d_1S_5) = 27. \end{aligned}$$

The numbers of rational places of $\mathcal{H}'_3$ over $\mathbb{F}_{2^4}$, $\mathbb{F}_{2^5}$ and $\mathbb{F}_{2^6}$ are respectively

$$\begin{aligned} N_4(\mathcal{H}'_3) &= 2^4 + 1 - S_4 = 14 \\ N_5(\mathcal{H}'_3) &= 2^5 + 1 - S_5 = 44 \\ N_6(\mathcal{H}'_3) &= 2^6 + 1 - S_6 = 38. \end{aligned}$$

We now have all the ingredients for the calculation of $L(T, \mathcal{H}'_4/\mathbb{F}_2)$. Following (6.7), let $\tilde{S}_r = N_r(\mathcal{H}'_3) - N_r(\mathcal{H}'_4)$ for $r = 1, 2, \ldots, 6$. We have

$$\begin{aligned} \tilde{S}_1 &= \tilde{S}_3 = \tilde{S}_5 = 0, \\ \tilde{S}_2 &= \tilde{S}_4 = -10, \quad \tilde{S}_6 = 14. \end{aligned}$$

By Newton's identity and (6.5),

$$\begin{aligned} \frac{L(T, \mathcal{H}'_4/\mathbb{F}_2)}{L(T, \mathcal{H}'_3/\mathbb{F}_2)} &= 1 + 5T^2 + 15T^4 + 31T^6 + 60T^8 + 80T^{10} + 64T^{12} \\ &= (1 + T + 2T^2)^2(1 - T + 2T^2)^2(1 - T^2 + 4T^4). \end{aligned}$$

Finally, by (6.6) we obtain

$$\frac{L(T, \mathcal{H}_4/\mathbb{F}_4)}{L(T, \mathcal{H}_3/\mathbb{F}_4)} = (1 + 3T + 4T^2)^4(1 - T + 4T^2)^2,$$

and therefore

$$L(T, \mathcal{H}_4/\mathbb{F}_4) = (1 + 3T + 4T^2)^7(1 - T + 4T^2)^2.$$

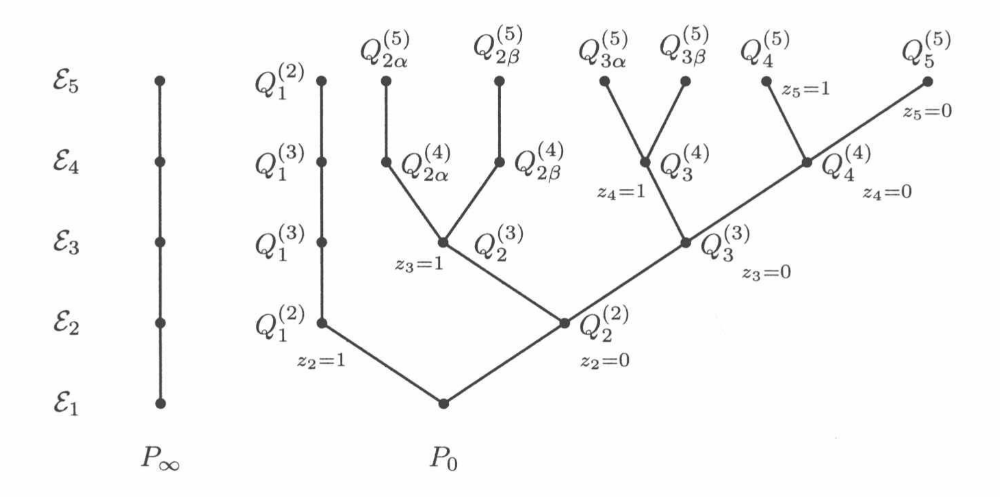



**Figure 1** Ramified places in GS1.

## 6.5 Conclusion

The two Garcia-Stichtenoth towers of function fields are explicit towers that attain the Drinfel-Vlăduţ bound. We presented some computational tricks and computed the zeta functions of the first few levels of the two Garica-Stichtenoth towers.

## Appendix: Counting points over $P_0$ in GS1

Recall that $P_0$ stands for the unique zero of $x_1$ in $\mathcal{E}_1 = \mathbb{F}_{q^2}(x_1)$. In this appendix, we will count the places above $P_0$ in $\mathcal{E}_i/\mathbb{F}_4$, for $i = 2, \ldots, 5$, and determine their degrees. The places lying above $P_0$ are depicted in Fig. 1.

We first fix some notations. For a place $P$ in function field $F$, we denote the normalized discrete valuation and the discrete valuation ring associated with $P$ by $v_P$ and $O_P$ respectively. For element $x$, $y$ and $z$ in $F$, we write

$$x = y + \mathcal{O}(z) \text{ at } P,$$

if $x = y + tz$ with $v_P(t) \geq 0$. The residue class map with respect to $P$ is denoted by $x \mapsto x(P)$. For a polynomial $g(T)$ with coefficients in $O_P$, we let $\bar{g}(T)$ be the polynomial with each coefficient replaced by its residue class in $O_P/P$.

**First level $\mathcal{E}_1/\mathbb{F}_4$.** By definition, $P_0$ is the unique zero of $x_1$ in the rational function field $\mathcal{E}_1 = \mathbb{F}_4(x_1)$. $x_1$ is a uniformizer of $P_0$.

**Second level $\mathcal{E}_2/\mathbb{F}_4$.** At all places lying above $P_0$, $x_1$ reduces to 0 under the residue class map. The defining polynomial $\phi_1(T) = T^2 + T - x_1^3$ of $\mathcal{E}_2$, is reduced to $\bar{\phi}_1(T) = T^2 + T$. There are two solutions of $\bar{\phi}_1(T) = 0$ in $\mathbb{F}_4$, namely 0 and 1. By Kummer's theorem [81, III.3.7], there are two rational places lying above $P_0$ in $\mathcal{E}_2$. We denote the place corresponding to the solution $T = 1$ by $Q_1^{(2)}$ and the place corresponding to $T = 0$ by $Q_2^{(2)}$. We will use superscript $^{(k)}$ to indicate that a place is in $\mathcal{E}_k$. It is shown in [31] that $Q_1^{(2)}$ will be totally ramified as we go up the tower, i.e., there is exactly one rational place lying above $Q_1^{(2)}$ in $\mathcal{E}_i$ for $i \geq 3$.

**Third level $\mathcal{E}_3/\mathbb{F}_4$.** At any place lying above $Q_2^{(2)}$, we have $v_{P_0}(z_2) > 0$ and $v_{P_0}(x_1) > 0$. From the equation $z_2^2 + z_2 = x_1^3$ and the strict triangular inequality, we have $v_{P_0}(z_2) = 3v_{P_0}(x_1) = 3$. Hence,

$$v_{P_0}(x_2) = v_{P_0}(z_2/x_1) = 2.$$

The defining polynomial $\phi_2(T) = T^2 + T - x_2^3$ of $\mathcal{E}_3$ is reduced to $\bar{\phi}_2(T) = T^2 + T$ via the residue class map of $Q_2^{(2)}$. As $\bar{\phi}_2(T) = 0$ has two solutions in $\mathbb{F}_4$, by Kummer's theorem, there are two places in $\mathcal{E}_3$ which lie above $Q_2^{(2)}$. We let $Q_2^{(3)}$ be the place corresponding to $T = 1$, and $Q_3^{(3)}$ be the place corresponding to $T = 0$.

There are three rational places in $\mathcal{E}_3$ above $P_0$.

**Fourth level $\mathcal{E}_3/\mathbb{F}_4$.** We first consider $Q_2^{(3)}$. We will show that there are two rational places over $\mathbb{F}_4$ lying above $Q_2^{(3)}$. To this end, we will express all variables $z_4$, $z_3$ etc. in terms of $x_1$, which is a uniformizer of $Q_2^{(3)}$.

From the defining equation, we get

$$\begin{aligned} z_2 &= x_1^3 + z_2^2 && (6.10)\\ &= x_1^3 + (x_1^3 + z_2^2)^2 && (6.11)\\ &= x_1^3 + x_1^6 + z_2^4. && (6.12)\end{aligned}$$

In the second equality above, we have substituted $z_2$ on the right hand side by $x_1^3 + z_2^2$. Repeating this procedure we obtain

$$z_2 = x_1^3 + x_1^6 + x_1^{12} + x_1^{24} + z_2^{16}. \qquad (6.13)$$

Thus,

$$z_2 = x_1^3 + x_1^6 + x_1^{12} + \mathcal{O}(x_1^{24}). \qquad (6.14)$$

Putting this in the definition of $x_2$, we have

$$x_2 = z_2/x_1 = x_1^2 + x_1^5 + x_1^{11} + \mathcal{O}(x_1^{23}). \qquad (6.15)$$

Since the residue $z_3(Q_2^{(3)})$ equals 1, we make a substitution $\tilde{z}_3 := z_3 - 1$. Then $\tilde{z}_3(Q_2^{(3)})$ and $v_{Q_2^{(3)}}(\tilde{z}_3) > 0$. By the similar calculation as from (6.10) to (6.14), we get

$$z_3 = 1 + \tilde{z}_3 = 1 + x_2^3 + x_2^6 + x_2^{12} + \mathcal{O}(x_2^{24}).$$

Therefore,

$$x_3 = \frac{1}{x_2} + x_2^2 + x_2^5 + x_2^{11} + \mathcal{O}(x_2^{23}),$$

and after taking the cube,

$$x_3^3 = \frac{1}{x_2^3} + 1 + x_2^3 + x_2^8 + x_2^{12} + \mathcal{O}(x_2^{15}).$$

Substitute $x_2$ by $x_1$ in the above equation using (6.15), and after some calculation, we obtain

$$x_3^3 = \frac{1}{x_1^6} + \frac{1}{x_1^3} + 1 + x_1^3 + \mathcal{O}(x_1^9). \qquad (6.16)$$

We now define a new variable $\tilde{z}_4 := z_4 + 1/x_1^3$. It is clear that $\mathcal{E}_4 = \mathcal{E}_3(z_4) = \mathcal{E}_3(\tilde{z}_4)$, and we can generate $\mathcal{E}_4$ using $\tilde{z}_4$ instead of $z_4$. From the definition,

$$z_4^2 + z_4 = x_3^3$$

and (6.16), we see that $\tilde{z}_4$ satisfies

$$\tilde{z}_4^2 + \tilde{z}_4 = 1 + x_1^3 + \mathcal{O}(x_1^9).$$

The polynomial $\phi_3(T) := T^2+T+1+x_1^3+\mathcal{O}(x_1^9)$ is reduced to $\bar{\phi}_3(T) = T^2+T+1$. The two roots of $\bar{\phi}_3(T)$ are the two elements in $\mathbb{F}_4 \setminus \mathbb{F}_2$. Let the two roots of $T^2+T+1$ be $\alpha$ and $\beta$. By Kummer's theorem, we conclude that there are two rational places over $\mathbb{F}_4$ that lies above $Q_2^{(3)}$, one associated to $\alpha$ and the other associated to $\beta$. We denote these two places by $Q_{2\alpha}^{(4)}$ and $Q_{2\beta}^{(4)}$.

Similar to $Q_2^{(2)}$ in $\mathcal{E}_2$, there are two rational places, $Q_3^{(4)}$ and $Q_4^{(4)}$, lying above $Q_3^{(3)}$, corresponding to the two solutions of $T^2+T=0$.

Therefore, in $\mathcal{E}_4/\mathbb{F}_4$, there are 5 rational places. If we consider $\mathcal{E}_4/\mathbb{F}_2$ as function field over $\mathbb{F}_2$, then $Q_{2\alpha}^{(4)}$ and $Q_{2\beta}^{(4)}$ will collapse into a single place of degree 2 over $\mathbb{F}_2$. For $\mathcal{E}_4/\mathbb{F}_2$, there are 3 rational places and 1 place of degree 2 over $P_0$.

**Fifth level $\mathcal{E}_5/\mathbb{F}_4$.** In the extension $\mathcal{E}_5/\mathcal{E}_4$, the two places $Q_{2\alpha}^{(4)}$ and $Q_{2\beta}^{(4)}$ are totally ramified [31]. There is precisely one place lying above each of them.

By the same argument for the extension of $Q_2^{(3)}$ in $\mathcal{E}_4/\mathcal{E}_3$, the place $Q_3^{(4)}$ splits into two places $Q_{3\alpha}^{(5)}$ and $Q_{3\beta}^{(5)}$ in the extension $\mathcal{E}_5/\mathcal{E}_4$, corresponding to the roots of $T^2+T+1$ in $\mathbb{F}_4$.

$Q_4^{(4)}$ splits into two places $Q_4^{(5)}$ and $Q_5^{(5)}$, corresponding to the roots of $T^2+T$ in $\mathbb{F}_4$.

There are 7 rational places in $\mathcal{E}_5/\mathbb{F}_4$ over $P_0$ in total. If constant field reduction is applied, then $\{Q_{2\alpha}^{(5)}, Q_{2\alpha}^{(5)}\}$ and $\{Q_{3\alpha}^{(5)}, Q_{3\alpha}^{(5)}\}$ will be two Galois orbits under the Galois group of $\mathbb{F}_4/\mathbb{F}_2$. Hence there are there are 3 rational places and 2 places of degree 2 in $\mathcal{E}_5/\mathbb{F}_2$ lying above $P_0$.

**Kenneth W. Shum**
Department of Information Engineering
The Chinese University of Hong Kong
Shatin, N.T. Hong Kong
wkshum@ie.cuhk.edu.hk

# Bibliography


[1] A. Akbary and V. Kumar Murty, Reduction mod $p$ of subgroups of the Mordell-Weil group of an elliptic curve, *Intl. J. Number Theory*, **5**(2009), 465–487.

[2] A. Akbary, D. Ghioca and V. Kumar Murty, Reduction of points on elliptic curves, *Math. Annalen.* **347**(2010), 365–394.

[3] I. Aleshnikov, P. V. Kumar, K. W. Shum, and H. Stichtenoth. On the splitting of places in a tower of function fields meeting Drinfeld-Vlăduţ bound. *IEEE Trans. Inform. Theory*, **47**(4)(2001), 1613–1619.

[4] S. Arita, *Algorithms for computations in Jacobian group of $C_{ab}$ curve and their application to discrete-log based public key cryptosystems*, Conference on the Mathematics of Public-Key Cryptography, June 12-17, 1999.

[5] S. Arita, An addition algorithm in Jacobian of $C_{ab}$ curves, *Discrete Appl. Math.*, **130**(2003), 13–31.

[6] S. Arita, S. Miura, and T. Sekiguchi, An addition algorithm on the Jacobian varieties of curves, *J. Ramanujan Math. Soc.*, **19**(2004), 235–251.

[7] Y. Aubry and M. Perret, Divisibility of zeta functions of curves in a covering. *Arch. Math.*, **82**(2004), 205–213.

[8] P. Berthelot, Géometrie rigide et cohomologie des variétés algébriques de characterisque $p$, Journées d'analyse $p$–adique, In: Introduction aux cohomologies $p$–adiques, *Bull. Soc. Math. France.*, **23**(1986), 7–32.

[9] P. Berthelot, Cohomologie rigide et cohomologie rigide à supports propre, Première partie (version provsoire 1991), Prépublication IRMAR 96-03, Université de Rennes (1996).

[10] P. Berthelot, Finitude et pureté cohomologique en cohomologie rigid (with an appendix in English by Aise Johan de Jong), *Invent. Math.* **128**(1997), 329–377.

[11] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, Cambridge, 1999.

[12] D. Boneh and R. Venkatesan, Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes, In: *CRYPTO 1996*, N. Koblitz (ed.), pp. 129–142, *Lecture Notes in Computer Science*, **1109**, Springer-Verlag, Berlin, 1996.

[13] J. Bourgain, New bounds on exponential sums related to the Diffie-Hellman distributions, *Comptes Rendus Acad. Sci. Paris*, **338**(2004), 825–830.

[14] B. Buchberger, Groebner bases: an algorithmic method in polynomial ideal theory, In: *Multidimensional Systems Theory*, N. K. Bose (ed.), pp. 184–232, D. Reidel Publishing Company, Dodrecht 1985.

[15] J. Buchmann and A. Schmidt, Computing the structure of a finite Abelian group, *Math. Comp.*, **74**(2005), 2017–2026.

[16] R. Canetti, J. Friedlander, S. Konyagin, M. Larsen, D. Lieman, and I. Shparlinski, On the statistical properties of Diffie-Hellman distributions, *Israel J. Math.*, **120**(2000), 23–46.

[17] D. G. Cantor, Computing in the Jacobian of hyperelliptic curve, *Math. Comp.*, **48**(1987), 95–101.

[18] J. Chao, K. Matsuo, and S. Tsujii, An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields, In: *Algorithmic Number Theory*, C. Fieker and D. R. Kohel (eds.), pp. 461–474, *Lecture Notes in Computer Science*, **2369**, Springer-Verlag, Berlin, 2002.

[19] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman and Hall/CRC, Boca Raton, 2006.

[20] A. Cojocaru, Reduction of an elliptic curve with almost prime orders, *Acta Arith.*, **119**(2005), 26–289.

[21] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms,* Springer-Verlag, Berlin, 1992.

[22] J. Denef and F. Vercauteren, An Extension of Kedlaya's Algorithm to Artin-Schreier curves in characteristic 2. Available at `http://www.wis.kuleuven.ac.be/algebra/denef.html`. 2002.

[23] J. Denef and F. Vercauteren, An Extension of Kedlaya's Algorithm to Hyperelliptic Curves in characteristic 2, *J. Cryptology*, **19**(2006), 1–25.

[24] C. Diem and E. Thomé, Index calculus in class groups of non-hyperelliptic curves of genus 3, *J. Cryptology*, **21**(2008), 593–611.

[25] D. Eisenbud, *Commutative Algebra with a View Towards Algebraic Geometry*, Springer-Verlag, Berlin, 1995.

[26] P. Erdös and M. Ram Murty, On the order of $a \pmod p$, In: *CRM Proceedings and Lecture Notes*, R. Gupta and K.S. Williams (eds.), pp. 87–97, **19**, American Mathematical Society, New York, 1999.

[27] M. Fouquet, P. Gaudry, and R. Harley, An extension of Satoh's algorithm and its implementation, *J. Ramanujan Math. Soc.*, **15**(2000), 281–318.

[28] M. Fouquet and F. Morain, Isogeny volcanoes and the SEA algorithm, In: *Proceedings of the 5th International Symposium on Algorithmic Number Theory,* C. Fieker and D. R. Kohel (eds.), pp. 276–291, *Lecture Notes in Computer Science*, **2369**, Springer-Verlag, London, UK, 2002.

[29] R. Froberg, *An Introduction to Gröbner Bases*, John Wiley, 1997.

[30] S. Galbraith and J. McKee, The probability that the number of points on an elliptic curve over a finite field is prime, *J. London Math. Soc.*, **62**(2000), 671–684.

[31] A. Garcia and H. Stichtenoth, A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlăduţ bound, *Inventiones Math.*, **121**(1995), 211–222.

[32] A. Garcia and H. Stichtenoth, On the asymptotic behaviour of some towers of function fields over finite fields, *J. of Number Theory*, **61**(1996), 248–273.

[33] P. Gaudry, An algorithm for solving the discrete log problem on hyperelliptic curves, In: *EUROCRYPT'00*, B. Preneel (ed.), pp. 19–34, *Lecture Notes in Computer Science*, **1807**, Springer-Verlag, Berlin, 2000.

[34] P. Gaudry, *Algorithms for counting points on curves,* Talk at ECC 2001, The Fifth Workshop on Elliptic Curves Cryptography, University of Waterloo, 2001.

[35] P. Gaudry and R. Harley, Counting points on hyperelliptic curves over finite fields, In: *Algorithmic Number Theory,* W. Bosma (ed.), pp. 313–332, *Lecture Notes in Computer Science*, **1838**, Springer-Verlag, Berlin, 2000.

[36] P. Gaudry, E. Thomé, N. Thériault, and C. Diem, A double large prime variation for small genus hyperelliptic index calculus, *Math. Comp.*, **76**(2007), 475–492.

[37] R. Gerkmann, Relative rigid cohomology and point counting on families of elliptic curves, *J. Ramanujan Math. Soc.*, **23**(2008), 1–31.

[38] R. Gerkmann, Extracts from his doctoral thesis (private communication), IEM Essen, Germany, 2003.

[39] R. Gupta and M. Ram Murty, Primitive points on elliptic curves, *Compositio Math.*, **58**(1986), 13–44.

[40] C. Guyot, K. Kaveh, and V. Patankar, Explicit algorithm for the arithmetic on the hyperelliptic Jacobians of genus 3, *J. Ramanujan Math. Soc.*, **19**(2004), 75–115.

[41] R. Harley, J. F. Mestre, and P. Gaudry, *Counting points with the arithmetic-geometric mean,* Rump talk at Eurocrypt 2001.

[42] R. Hartshorne, *Algebraic Geometry,* Springer-Verlag, Berlin, 1997.

[43] F. Hess, Computing Riemann-Roch spaces in algebraic function fields and related topics, *J. Symbolic Comput.*, **33**(2002), 425–445.

[44] T. Hoholdt, J. H. van Lint, and R. Pellikaan, Order functions and evaluation codes, In: *Applied Algebra and Error Correcting Codes-12*, T. Mora and H. Mattson (eds.), pp. 138–150, *Lecture Notes in Computer Science*, **1255**, Springer-Verlag, Berlin, 1997.

[45] F. Izadi and V. Kumar Murty, Counting points on an Abelian variety over a finite field, In: *INDOCRYPT 2003*, G. Goos, J. Hartmanis, and J. van Leeuwen (eds.), pp. 323–333, *Lecture Notes in Computer Science*, **2904**, Springer-Verlag, Berlin, 2003.

[46] M. Jacobson, A. Menezes, and A. Stein, Solving elliptic curve discrete logarithm problems using Weil descent, *J. Ramanujan Math. Soc.*, **16**(2001), 231–260.

[47] N. Jacobson *Basic Algebra-II*, W.H. Freeman and Company, 1980.

[48] K. S. Kedlaya, Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology, *J. Ramanujam Math. Soc.*, **16**(2001), 323–338.

[49] S. L. Kleiman, Algebraic cycles and the Weil conjectures, In: *Dix Exposés sur la Cohomolgie des Schémas*, pp. 359–386, North-Holland, Amsterdam; Masson, Paris, 1968.

[50] N. Koblitz, Primality of the number of points on an elliptic curve over a finite field, *Pacific J. Math.*, **31**(1988), 157–165.

[51] N. Koblitz, Hyperelliptic cryptosystems, *J. Cryptography*, **1**(1989), 139–150.

[52] N. Koblitz, A very easy way to generate curves over prime fields for hyperelliptic cryptosystems, RUMP Talk, Crypto 1997.

[53] K. Khuri-Makdisi, Asymptotically fast group operations on Jacobians of general curves. *Math. Comp.*, **76**(2007), 2213–2239.

[54] S. Lang, *Elliptic Curves Diophantine Analysis*, Springer-Verlag, Berlin, 1978.

[55] T. Lange, Formulae for arithmetic on genus 2 hyperelliptic curves, In: *Applied Algebra and Error Correcting Codes*, M. Fossorier, T. Hoholdt, and A. Poli (eds.), pp. 295–328, *Lecture Notes in Computer Science*, **2643**, Springer-Verlag, Berlin, 2003.

[56] T. Lange and I. Shparlinski, Distribution of some sequences of points on elliptic curves, *J. Math. Crypt.*, **1**(2007), 1–11.

[57] A. G. B. Lauder and D. Wan, Counting points on varieties over finite fields of small characteristic, In: *Algorithmic Number Theory*, J. Buhler and P. Stevenhagen (eds.), pp. 579–612, Cambridge University Press, Cambridge, 2008.

[58] A. G. B. Lauder and D. Wan, Computing zeta functions of Artin-Schreier curves over finite fields, *London Math. Soc. JCM*, **5**(2002), 34–55.

[59] Y.-R. Liu, Prime divisors of the number of rational points on elliptic curves with complex multiplication, *Bull. London Math. Soc.*, **37**(2005), 658–664.

[60] N. Matsuda, J. Chao, and S. Tsujii, Efficient construction algorithms of secure hyperelliptic discrete logarithm problems, IEICE ISEC96-18(1996).

[61] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[62] A. Miola and T. Mora, Constructive lifting in graded structures: a unified view of Buchberger and Hensel methods, *J. Symbolic. Comput.*, **6**(1988), 305–322.

[63] S. A. Miri and V. Kumar Murty, An application of sieve methods to elliptic curves, In: *INDOCRYPT 2001*, C. Pandu Rangan and C. Ding (eds.), pp. 91–98, *Lecture Notes in Computer Science*, **2247**, Springer-Verlag, Berlin, 2001.

[64] S. Miura and N. Kamiya, Geometric Goppa codes on some maximal curves and their minimum distance, In: *Proc. IEEE Workshop on Information Theory*, pp. 85–86, Susono-shi, Japan, June 1993.

[65] P. Monsky, Formal cohomology II: The cohomology sequence of a pair, *Ann. of Math.*, **88**(1968), 218–238.

[66] P. Monsky, Formal cohomology III: Fixed point theorems, *Ann. of Math.*, **93**(1971), 315–343.

[67] P. Monsky, Formal cohomology I, *Ann. of Math.*, **88**(1968), 181–217.

[68] F. Morain, Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques, *Journal de Théorie des Nombres de Bordeaux*, **7**(1995), 255–282.

[69] V. Kumar Murty and V. Patankar, Splitting of Abelian varieties, *Intl. Math. Res. Not.*, **2008**(2008), doi:10.1093/imrn/rn033.

[70] H. Niederrieter and C. Xing, Rational points on curves over finite fields, In: *London Mathematical Society Lecture Note Series* , **285**, Cambridge University Press, Cambridge, 2001.

[71] P. Ribenboim, Equivalent forms of Hensel's lemma, *Exposition. Math.* **3**(1985), 3–24.

[72] T. Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting, *J. Ramanujan Math. Soc.*, **15**(2000), 247–270.

[73] T. Satoh and K. Araki, Fermat quotients and the polynomial time Discrete log algorithm for anomalous elliptic curves, In: *Commentarii Mathematici Universitatis Sancti Pauli*, **47**(1998), 81–92.

[74] R. Schoof, Elliptic curves over finite fields and the computation of square roots mod $p$, *Math. Comp.*, **44**(1985), 483–494.

[75] R. Schoof, Counting points on elliptic curves over finite fields, *Journal de Théorie des Nombres de Bordeaux*, **7**(1995), 219–254.

[76] J. H. Silverman, *The Arithmetic of Elliptic Curves,* Springer-Verlag, Berlin, 1986.

[77] C. Sims, Computing the order of a solvable permutation group, *J. Symbolic Comp.*, **9**(1990), 699–705.

[78] B. Smith, Isogenies and the discrete logarithm problem in Jacobians of genus 3 hyperelliptic curves, In: *EUROCRYPT 2008*, N. Smart (ed.), pp. 163–180, *Lecture Notes in Computer Science*, **4965**, Springer-Verlag, Berlin, 2008.

[79] B. Smith, Isogenies and the discrete logarithm problem in Jacobians of genus 3 hyperelliptic curves (Extended version), *J. Cryptology*, **22**(2009), 505–529.

[80] A. M. Spallek, *Kurven vom geshlecht 2 und ihre anwendung in public-key-kryptosystemen,* Doctor thesis, Universitat GH Essen, 1994.

[81] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer-Verlag, Berlin, 1993.

[82] N. Thériault, Index calculus attack for hyperelliptic curves of small genus, In: *ASIACRYPT 2003*, C. S. Laih (ed.), pp. 75–92, *Lecture Notes in Computer Science*, **2894**, Springer-Verlag, Berlin, 2003.

[83] M. A. Tsfasman, S. G. Vlăduţ, and T. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound, *Math. Nachr.*, **109**(1982), 21–28.

[84] M. van der Put, The cohomology of Monsky and Washnitzer, *Mém. Soc. Math. France,* **22**(1986), 33–60.

[85] E. J. Volcheck, Computing in the Jacobian of a plane algebraic curve, In: *Proceedings of the First International Symposium on Algorithmic Number Theory*, L. M. Adleman and M-D. A. Huang (eds.), pp. 221–233, *Lecture Notes in Computer Science*, **877**, Springer-Verlag, Berlin, 1994.

[86] J. von zur Gathen, Hensel and Newton methods in valuation rings, *Math. Comp.*, **42**(1984), 637–661.

[87] D. Wan, Computing zeta functions over finite fields, *Contemporary Mathematics*, **225**(1999), 131–141.

[88] T. Wollinger, J. Pelzl, and C. Paar, Cantor versus Harley: Optimization and analysis of explicite formulae for hyperelliptic curve cryptosystems, *IEEE Transactions on Computers*, **54**(2005), 861–872.

[89] A. Zaytsev and G. McGuire, On the zeta functions of an optimal tower of function fields over $\mathbb{F}_4$, preprint, arXiv:0906.5252v1, 2009.

# Index

反盗版举报电话 (010) 58581999 58582371 58582488
反盗版举报传真 (010) 82086060
反盗版举报邮箱 dd@hep.com.cn
通信地址 北京市西城区德外大街4号
高等教育出版社法律事务与版权管理部
邮政编码 100120

# 美国数学会经典影印系列

| No. | Title | ISBN |
|---|---|---|
| 1 | **Lars V. Ahlfors**, Lectures on Quasiconformal Mappings, Second Edition | 9787040470109 |
| 2 | **Dmitri Burago, Yuri Burago, Sergei Ivanov**, A Course in Metric Geometry | 9787040469080 |
| 3 | **Tobias Holck Colding, William P. Minicozzi II**, A Course in Minimal Surfaces | 9787040469110 |
| 4 | **Javier Duoandikoetxea**, Fourier Analysis | 9787040469011 |
| 5 | **John P. D'Angelo**, An Introduction to Complex Analysis and Geometry | 9787040469981 |
| 6 | **Y. Eliashberg, N. Mishachev**, Introduction to the *h*-Principle | 9787040469028 |
| 7 | **Lawrence C. Evans**, Partial Differential Equations, Second Edition | 9787040469356 |
| 8 | **Robert E. Greene, Steven G. Krantz**, Function Theory of One Complex Variable, Third Edition | 9787040469073 |
| 9 | **Thomas A. Ivey, J. M. Landsberg**, Cartan for Beginners: Differential Geometry via Moving Frames and Exterior Differential Systems | 9787040469172 |
| 10 | **Jens Carsten Jantzen**, Representations of Algebraic Groups, Second Edition | 9787040470086 |
| 11 | **A. A. Kirillov**, Lectures on the Orbit Method | 9787040469103 |
| 12 | **Jean-Marie De Koninck, Armel Mercier**, 1001 Problems in Classical Number Theory | 9787040469998 |
| 13 | **Peter D. Lax, Lawrence Zalcman**, Complex Proofs of Real Theorems | 9787040470000 |
| 14 | **David A. Levin, Yuval Peres, Elizabeth L. Wilmer**, Markov Chains and Mixing Times | 9787040469943 |
| 15 | **Dusa McDuff, Dietmar Salamon**, *J*-holomorphic Curves and Symplectic Topology | 9787040469936 |
| 16 | **John von Neumann**, Invariant Measures | 9787040469974 |
| 17 | **R. Clark Robinson**, An Introduction to Dynamical Systems: Continuous and Discrete, Second Edition | 9787040470093 |
| 18 | **Terence Tao**, An Epsilon of Room, I: Real Analysis: pages from year three of a mathematical blog | 9787040469004 |
| 19 | **Terence Tao**, An Epsilon of Room, II: pages from year three of a mathematical blog | 9787040468991 |
| 20 | **Terence Tao**, An Introduction to Measure Theory | 9787040469059 |
| 21 | **Terence Tao**, Higher Order Fourier Analysis | 9787040469097 |
| 22 | **Terence Tao**, Poincaré's Legacies, Part I: pages from year two of a mathematical blog | 9787040469950 |
| 23 | **Terence Tao**, Poincaré's Legacies, Part II: pages from year two of a mathematical blog | 9787040469967 |
| 24 | **Cédric Villani**, Topics in Optimal Transportation | 9787040469219 |
| 25 | **R. J. Williams**, Introduction to the Mathematics of Finance | 9787040469127 |
| 26 | **T. Y. Lam**, Introduction to Quadratic Forms over Fields | 9787040469196 |

27 **Jens Carsten Jantzen**, Lectures on Quantum Groups — 9 787040 469141

28 **Henryk Iwaniec**, Topics in Classical Automorphic Forms — 9 787040 469134

29 **Sigurdur Helgason**, Differential Geometry, Lie Groups, and Symmetric Spaces — 9 787040 469165

30 **John B.Conway**, A Course in Operator Theory — 9 787040 469158

31 **James E. Humphreys**, Representations of Semisimple Lie Algebras in the BGG Category *O* — 9 787040 468984

32 **Nathanial P. Brown, Narutaka Ozawa**, C*-Algebras and Finite-Dimensional Approximations — 9 787040 469325

33 **Hiraku Nakajima**, Lectures on Hilbert Schemes of Points on Surfaces — 9 787040 501216

34 **S. P. Novikov, I. A. Taimanov, Translated by Dmitry Chibisov**, Modern Geometric Structures and Fields — 9 787040 469189

35 **Luis Caffarelli, Sandro Salsa**, A Geometric Approach to Free Boundary Problems — 9 787040 469202

36 **Paul H. Rabinowitz**, Minimax Methods in Critical Point Theory with Applications to Differential Equations — 9 787040 502299

37 **Fan R. K. Chung**, Spectral Graph Theory — 9 787040 502305

38 **Susan Montgomery**, Hopf Algebras and Their Actions on Rings — 9 787040 502312

39 **C. T. C. Wall, Edited by A. A. Ranicki**, Surgery on Compact Manifolds, Second Edition — 9 787040 502329

40 **Frank Sottile**, Real Solutions to Equations from Geometry — 9 787040 501513

41 **Bernd Sturmfels**, Gröbner Bases and Convex Polytopes — 9 787040 503081

42 **Terence Tao**, Nonlinear Dispersive Equations: Local and Global Analysis — 9 787040 503050

43 **David A. Cox, John B. Little, Henry K. Schenck**, Toric Varieties — 9 787040 503098

44 **Luca Capogna, Carlos E. Kenig, Loredana Lanzani**, Harmonic Measure: Geometric and Analytic Points of View — 9 787040 503074

45 **Luis A. Caffarelli, Xavier Cabré**, Fully Nonlinear Elliptic Equations — 9 787040 503067

46 **Teresa Crespo, Zbigniew Hajto,** Algebraic Groups and Differential Galois Theory — 9 787040 510133

47 **Barbara Fantechi, Lothar Göttsche, Luc Illusie, Steven L. Kleiman, Nitin Nitsure, Angelo Vistoli,** Fundamental Algebraic Geometry: Grothendieck's FGA Explained — 9 787040 510126

48 **Shinichi Mochizuki,** Foundations of $p$-adic Teichmüller Theory — 9 787040 510089

49 **Manfred Leopold Einsiedler, David Alexandre Ellwood, Alex Eskin, Dmitry Kleinbock, Elon Lindenstrauss, Gregory Margulis, Stefano Marmi, Jean-Christophe Yoccoz,** Homogeneous Flows, Moduli Spaces and Arithmetic — 9 787040 510096

50 **David A. Ellwood, Emma Previato,** Grassmannians, Moduli Spaces and Vector Bundles — 9 787040 510393

51 **Jeffery McNeal, Mircea Mustaţă,** Analytic and Algebraic Geometry: Common Problems, Different Methods — 9 787040 510553

52 **V. Kumar Murty,** Algebraic Curves and Cryptography — 9 787040 510386

53 **James Arthur, James W. Cogdell, Steve Gelbart, David Goldberg, Dinakar Ramakrishnan, Jiu-Kang Yu,** On Certain $L$-Functions — 9 787040 510409

本版只限于中华人民共和国境内发行。本版经由美国数学会授权仅在中华人民共和国境内销售，不得出口。

利用有限 Abel 群构建公钥密码系统现在已经成为著名的范例，而代数几何学通过有限域上的 Abel 簇提供了一些这样的群，特别令人感兴趣的是 Abel 簇为代数曲线的 Jacobi 簇的情形。本书中的所有文章都聚焦于有限域上曲线的 Jacobi 簇的点计数和显式算法这一主题。这些文章的论题包括 Schoof 的 $l$ 进点计数算法、Kedlaya 和 Denef-Vercauteren 的 $p$ 进算法、$C_{ab}$ 曲线和 zeta 函数的 Jacobi 簇的显式算法。

本书的文章大部分都适合希望进入这一领域的研究生独立学习，这些文章既介绍了基础性材料，又能引导读者深入到文献中去。密码学的文献看上去是呈指数型增长的，对于一个入门者来说，穿越这片海洋令人望而却步。本书会将读者引向关于这一数学分支的若干新思想的讨论，并给出进一步阅读的简明指引。

本书适合对密码学以及数论和代数几何的应用感兴趣的研究生和研究人员阅读。

美国数学会经典影印系列




ISBN 978-7-04-051038-6




定价 67.00 元